

Method and apparatus for device interaction by format

Publication number: JP2001524231 (T)

Publication date: 2001-11-27

Inventor(s):

Applicant(s):

Classification:

- International: G06F13/00; G06F13/38; G06F15/177; G06F3/12; G06F3/14; H04L29/06; H04L29/08; H04N1/32; H04N5/00; H04Q9/00; G06F13/00; G06F13/38; G06F15/16; G06F3/12; G06F3/14; H04L29/06; H04L29/08; H04N1/32; H04N5/00; H04Q9/00; (IPC1-7): G06F3/14; H04N5/00; H04Q9/00

- European: H04L29/06P; H04L29/08T; H04L29/08A7

Application number: JP19980543630T 19980415

Priority number(s): GB19970007551 19970415; WO1998GB01097 19980415; US19970054047P 19970718

Also published as:

EP0872991 (A2)
EP0872991 (A3)
EP0872991 (B1)
JP10320310 (A)
JP10304013 (A)

more >>

Abstract not available for JP 2001524231 (T)

Abstract of corresponding document: EP 0872991 (A2)

A method of passing information between two or more information handling devices is described. Such information handling devices might be a printer 342, a personal computer 343, or a scanner 344. Means for communication of information between information handling devices in the form of a network 341 and network connection means 345 exist. <??>The information transmitted comprises a data format hierarchy, wherein a device intended to receive transmitted data evaluates the data format hierarchy and determines the format in which the data is then received thereby.; <??>
>Advantageously, the receiving device determines the format in which the data is then received by a response to the transmitting device comprising a path through the data format hierarchy, and all data formats comprise one or more of a plurality of data format types, and wherein for each data format type, there exists a data format receivable by all information handling devices supporting that data format type. It is advantageous in implementation if the method of passing information further comprises requests for content data for a chosen path through the data format hierarchy and responses to such requests.

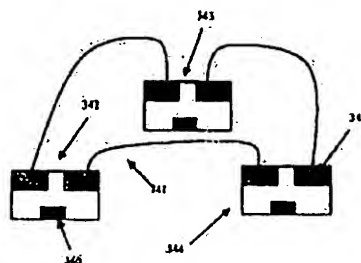


Figure 34

Data supplied from the *espacenet* database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号
特表2001-524231
(P2001-524231A)

(43) 公表日 平成13年11月27日 (2001. 11. 27)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 3/14	3 1 0	G 0 6 F 3/14	3 1 0 A
H 0 4 N 5/00		H 0 4 N 5/00	A
H 0 4 Q 9/00	3 0 1	H 0 4 Q 9/00	3 0 1 E
	3 3 1		3 3 1 A

審査請求 未請求 予備審査請求 有 (全 85 頁)

(21) 出願番号 特願平10-543630
(86) (22) 出願日 平成10年4月15日 (1998. 4. 15)
(85) 翻訳文提出日 平成11年10月8日 (1999. 10. 8)
(86) 国際出願番号 P C T / G B 9 8 / 0 1 0 9 7
(87) 国際公開番号 W O 9 8 / 4 7 0 7 6
(87) 国際公開日 平成10年10月22日 (1998. 10. 22)
(31) 優先権主張番号 9 7 0 7 5 5 1. 9
(32) 優先日 平成9年4月15日 (1997. 4. 15)
(33) 優先権主張国 イギリス (G B)
(31) 優先権主張番号 6 0 / 0 5 4, 0 4 7
(32) 優先日 平成9年7月18日 (1997. 7. 18)
(33) 優先権主張国 米国 (U S)

(71) 出願人 ヒューレット・パカード・カンパニー
アメリカ合衆国94304カリフォルニア州パ
ロアルトハノーバー・ストリート 3000
(72) 発明者 アーノルド・パトリック・シモン
アメリカ合衆国83706アイダホ州ボイジー、
サウス・ホワイト・バイン・プレイス
2152
(72) 発明者 メイサー・ピーター・ジェームズ
イギリス、ビーエス16、5アールエイチ、
プリストル、ダウンエンド、チェスターフ
ィールド・ロード 21
(74) 代理人 弁理士 岡田 次生

最終頁に続く

(54) 【発明の名称】 装置制御の方法および装置

(57) 【要約】

本発明の方法および装置において、1つの装置12が他の装置11を制御するために使用される。制御は、制御する装置12が制御される装置11の性質または機能性についていかなる知識をも持つことを必要としない。制御される装置11は、可能なパラメータ選択項目のセットを、可能なパラメータ選択項目を識別する情報と共に、制御する装置12へ伝送する。次に、制御する装置12は、ユーザ13が制御する装置12のユーザ・インターフェースを通して実際のパラメータ選択項目セットを選択することができるように、パラメータ選択項目セットおよび情報を表示する(またはユーザに提供する)。次に、実際のパラメータ選択項目セットが制御される装置11へ送り戻され、その実際のパラメータ選択項目セットに従って制御される装置11の動作パラメータが修正される。このように、制御する装置12は、制御される装置11の性質の知識を必要とせず、制御される装置自体によって提供される情報および選択項目をユーザ13に提示するだけでよい。この提示の性質は、制御する装置12の表示およびユーザ・インターフェース能力に

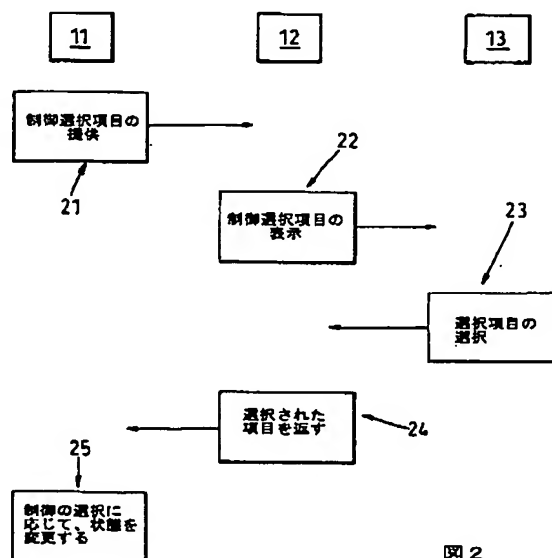


図 2

【特許請求の範囲】

1. 制御する装置による制御される装置の制御の方法であって、

制御する装置と制御される装置の間で信号を送送する手段を確立するステップと、

制御される装置が可能なパラメータ選択項目セットおよび上記可能なパラメータ選択項目を識別する情報を作成してそれらを制御する装置へ送送するステップと、

ユーザ選択手段を用いて実際のパラメータ選択項目セットをユーザが選択することができるように、制御する装置において上記可能なパラメータ選択項目セットおよび上記可能なパラメータ選択項目を識別する上記情報を表示するステップと、

制御される装置へ上記実際のパラメータ選択項目セットを送送するステップと、

上記実際のパラメータ選択項目セットに従って制御される装置の動作パラメータを修正するステップと、

を含み、

上記可能なパラメータ選択項目セットおよび上記実際のパラメータ選択項目セットが上記ユーザ選択手段から独立した形式で提供される、方法。

2. 実際パラメータ選択項目セットのユーザ選択、実際パラメータ選択項目セットの送送、および実際パラメータ選択項目セットに従った制御される装置の動作パラメータの修正という上記ステップが実際パラメータ選択項目の選択の修正をユーザが望む時はいつでも1つのシーケンスとして反復される、請求項1に記載の方法。

3. 上記可能なパラメータ選択項目セットが階層として構成され、所与のパラメータ選択項目の選択が、上記所与のパラメータ選択項目が選択されなかったならば必要とされない別のパラメータ選択項目のユーザ選択を必要とする、請求項1

または請求項2のいずれかに記載の方法。

4. 制御する装置と制御される装置の間で信号を送送する手段を確立する上記ス

ステップと、制御される装置が可能なパラメータ選択項目セットおよび上記可能なパラメータ選択項目を識別する情報を作成してそれらを制御する装置へ伝送する上記ステップの間に、制御される装置に対して上記可能なパラメータ選択項目セットおよび上記可能なパラメータ選択項目を識別する情報を制御する装置が要求するステップが含まれる、請求項1乃至請求項3のいずれかに記載の方法。

5. 上記可能なパラメータ選択項目セットおよびその可能なパラメータ選択項目に関連する情報が、各パラメータ毎に可能なパラメータ選択項目、そのパラメータのタイプおよびユーザが解釈可能なパラメータ情報を表す制御スクリプトの形式で提供される、請求項1乃至請求項4のいずれかに記載の方法。

6. 制御する装置が、あらかじめ定められた表示規則に従ってパラメータのタイプの各々を表示するように構成される、請求項5に記載の方法。

7. 上記可能なパラメータ選択項目セットおよび可能なパラメータ選択項目に関連する情報がマークアップ言語で提供される、請求項1乃至請求項4のいずれかに記載の方法。

8. 上記可能なパラメータ選択項目のセットおよび可能なパラメータ選択項目を識別している上記情報の制御する装置における表示が、ユーザに対する表示のため制御パネルを提示するステップを含み、そのためユーザが制御パネルと対話して、実際のパラメータ選択項目セットを提供することができる、請求項1乃至請求項7のいずれかに記載の方法。

9. 制御パネルにおける所与のパラメータ選択項目の表現が、そのパラメータの機能とは無関係であるが、そのパラメータに関して使用できる選択項目のタイプ

に依存する、請求項8に記載の方法。

10. 制御する装置においてパラメータ選択項目セットおよび可能なパラメータ選択項目を識別する情報を表示するステップが、上記可能なパラメータ選択項目セットおよび情報をその情報を解釈することができる手段が利用することができるようにするステップと、上記可能なパラメータ選択項目セットから選択された実際のパラメータ選択項目を返すステップを含む、請求項1乃至請求項7のいずれかに記載の方法。

11. 制御する装置と制御される装置の間の情報の各節が対話セットからの1つまたは複数の対話からなり、その対話セットのどの対話もその情報に対していずれかの装置によって実行される機能に依存しない、請求項1乃至請求項10のいずれかに記載の方法。

12. 情報の節の各々は装置間のサーフェスの共有に関連し、上記サーフェスが制御される装置の内部状態の表現である、請求項11に記載の方法。

13. 上記サーフェスの各々は、データおよび該データが制御される装置によって提供されることができる形式のセットを含む、請求項12に記載の方法。

14. 上記対話のセットが、ある1つのサーフェスに関してそのサーフェスを持っていない1つまたは複数の装置とのサーフェスの共有に関する対話を含む、請求項12または請求項13に記載の方法。

15. 上記対話セットが、第1の装置におけるサーフェスと1つまたは複数の第2の装置における対応するサーフェスの間での内容の伝送に関する対話を含む、請求項12乃至請求項14のいずれかに記載の方法。

16. 上記対話セットが、第1の装置におけるサーフェスに対応している1つまたは複数の第2の装置におけるサーフェスの削除または修正に関する対話を含む、請求項12乃至請求項15のいずれかに記載の方法。

17. サーフェスがサーフェスの機能を表すクラスを含む、請求項12乃至請求項16のいずれかに記載の方法。

18. 上記対話セットが第1の装置から第2の装置によるサーフェスに関する要求を含む、請求項12乃至請求項17のいずれかに記載の方法。

19. 上記対話セットが第1の装置から第2の装置によるサーフェスの記述に関する要求を含み、サーフェスの記述がそのサーフェスに関連するデータの作成のための選択項目の階層を含む、請求項12乃至請求項18のいずれかに記載の方法。

20. 上記対話セットがサーフェスに関連するデータの作成に関する要求を含む、請求項2乃至請求項19のいずれかに記載の方法。

21. 上記対話セットがサーフェスに関連するデータの作成に関する要求への応

答を含む、請求項20に記載の方法。

22. 上記対話セットがサーフェスへの変更を要求するメッセージを含む、請求項12乃至請求項21のいずれかに記載の方法。

23. サーフェスへの変更を要求する上記メッセージが、サーフェスへの変更を実行または無効にするためにも使用することができる、請求項22に記載の方法。

24. 制御する装置と制御される装置の間で交換される情報がデータ形式階層を含み、伝送されたデータを受け取るように意図される装置が、そのデータ形式階

層を評価してデータが受け取られる形式を決定する、請求項1乃至請求項23のいずれかに記載の方法。

25. データ形式階層を通過する経路を含む送信側装置への応答によってデータが受け取られる形式を受け取り側装置が決定する、請求項24に記載の方法。

26. すべてのデータ形式が複数のデータ形式タイプのうちの1つまたは複数を含み、各データ形式タイプに関して、そのデータ形式タイプをサポートするいかなる装置によってでも受け取られることができるデータ形式が存在する、請求項24または請求項25に記載の方法。

27. データ形式階層が、階層構造、伝送される情報の特性に関するキーワードのセットおよびキーワードの各々に関連する値を含む、請求項24乃至請求項26のいずれかに記載の方法。

28. 各データ形式階層が1つの符号化を含み、各符号化は、情報が提示されることができる異なる基本形式を表す、請求項27に記載の方法。

29. 制御する装置および制御される装置が情報処理装置である、請求項1乃至請求項28のいずれかに記載の方法。

30. 制御する装置および制御される装置が情報伝送ネットワークを経由して接続される、請求項29に記載の方法。

31. ユーザ選択手段がグラフィカル・ユーザ・インターフェースを含む、請求項1乃至請求項30のいずれかに記載の方法。

32. 請求項29の方法における制御する装置の機能に適応される情報処理装置

であって、

制御される装置との情報通信のための手段と、

ユーザに情報を表示してユーザ入力から取り出される値を返すユーザ・インターフェース手段と、

制御される装置から受け取られる可能なパラメータ選択項目セットおよび関連情報を上記ユーザ・インターフェース手段を通して提示し、上記ユーザ・インターフェース手段から得られる実際パラメータ選択項目を制御される装置へ返す手段と、

を備える情報処理措置。

33. 請求項29の方法における制御される装置の機能に適応される情報処理装置であって、

制御する装置との通信のための手段と、

可能なパラメータ選択項目のセットおよびの関連する情報を制御する装置へ提供し、実際のパラメータ選択項目のセットに従って該装置の動作パラメータの修正を行う手段と、

を備える情報処理装置。

34. 可能なパラメータ選択項目セットおよび上記可能なパラメータ選択項目を識別する情報の制御される装置による作成および制御する装置への伝送を可能にする制御スクリプトであって、各パラメータに関する可能なパラメータ選択項目、そのパラメータに関するタイプおよびユーザ解釈可能なパラメータ情報を含む、制御スクリプト。

35. 1つのパラメータのタイプが文字列セットである、請求項34に記載の制御スクリプト。

36. 1つのパラメータのタイプが数値の範囲である、請求項34または請求項35に記載の制御スクリプト。

37. 1つのパラメータのタイプが1つまたは複数の文字列および1つまたは複数の値である、請求項34または請求項35に記載の制御スクリプト。

38. 該制御スクリプトがパラメータの階層を確立する手段を更に含み、一定のパラメータ選択項目が他のパラメータ選択項目の結果に依存する、請求項34乃至請求項37のいずれかに記載の制御スクリプト。

【発明の詳細な説明】

装置制御の方法および装置

技術分野

本発明は、装置制御のための方法および装置に関するものであり、特に、ユーザが対話することができる別の装置を通して1つの装置を制御する方法およびその方法を活用する装置に関するものである。

背景技術

比較的便利でない対話機能を持つ別の装置を制御するため便利な対話機能を持つ装置をユーザが使用することは一般的である。例えば、テレビの制御のため遠隔制御装置が頻繁に使用される。遠隔制御装置はユーザの使用にとって便利である。すなわち、テレビ自体に装備されるユーザ・インターフェースが(テレビに固定されているので)使用上あまり便利でないのに対して、遠隔装置は携帯可能である。コンピュータも同様に他の装置を制御するためしばしば使用される。なぜならば、コンピュータのユーザ・インターフェースがユーザによってよく理解されていて、多くの場合(例えばデータが装置との間で送受信されるとすれば)装置とコンピュータの間の対話がいずれにせよ必要とされるからである。

異なるコンピュータ・オペレーティング・システムに関して異なるプログラミング・オプションでコンピュータによる装置の制御を行う種々の手法がある。例えば、マイクロソフト・ウインドウズにおいて、装置は、その装置クラスにとって適切とみなされる一組の制御コマンドがそのモデルの下で理解されなければならない定義済み共通モデル(そのためのAPIが提供される)に準拠するか、さもなければ、(制御が発生する前にコンピュータにロードされなければならない)当該装置特有の制御プログラムまたはドライバが供給されなければならない。APIのケースでは、例えば、TVAPIが、TVをオンまたはオフにし、特定のセットの値にチャンネルを変更し、あるいは音量をセットするような関数呼出しを持つことが可能である。この場合、すべてのテレビがこのような機能を装備して、APIを経由する制御のみを提供することができなければならない。いずれ

のケースにおいても、例えばパソコンによるファックス機の制御のためマイクロ

ソフト・ウインドウズが使用されるとすれば、ファックス機の機能性がパソコンによって”既知である”ことが必要である。ユーザ・イベントがパソコンによって検出され、次にファックス機によって必要とされる制御の文脈においてそれが解読される。

もう1つの例はXウインドウズによって与えられる。これは、典型的にはUnix上に構築されたGUIであって、別の装置の制御のため使用されることができる。Xウインドウズはマウスやキーボードのような種々の入力装置を認識する。それは、それら入力装置に何が起きているか(例えば座標345,678で右マウスボタンが押された)を記述するイベントをアプリケーションに提供するように適応される。次に、アプリケーションはこの動作をどのように解釈するかを決定する(例えばテレビをオンにする)。従って、制御プログラムは、制御される装置の機能性と制御する装置のユーザ・インターフェース機能の両方を必要とする。

そのような制御手法は、非常に強力であるとはいえ、柔軟でない。離散的な制御プログラムがあらゆる装置の組み合わせのため(またはウインドウズ・オペレーティング・システムの1バージョンを実行しているパーソナル・コンピュータのような1つのクラスの制御側に対する制御される装置に関していかなる率においても)工夫されなければならない。あるいは、装置の機能性が装置のタイプのあらかじめ定められた共通モデルに準拠しなければならない。従って、新しい装置のための制御プログラムを工夫し、所与の装置組み合わせに対して達成することができる最高水準の効率を維持する際に、大幅な再作業を行うことなく、完全な機能を備えた制御を実現できる一層柔軟な制御方法を達成することが望まれている。

発明の開示

本発明は、制御する装置が制御される装置の特定の知識を持たなければならない必要性を取り除くことによって上述の目標を達成することができることを認識する。

従って、本発明は、第1の側面において、制御する装置による制御される装置の制御の方法を提供する。該方法は、制御する装置と制御される装置の間で信号

を伝送する手段の確立、制御される装置における可能なパラメータ選択項目セットおよび可能なパラメータ選択項目を識別する情報の作成および制御する装置へのそれらの伝送、ユーザ選択手段を使用して実際のパラメータ選択項目セットのユーザ選択を可能にするため制御する装置における上記可能なパラメータ選択項目セットおよび当該可能なパラメータ選択項目セットを識別する情報の表示、および、上記実際のパラメータ選択項目セットに従った制御される装置の動作パラメータの修正を含み、該方法において、上記可能なパラメータ選択項目セットおよび実際のパラメータ選択項目セットが上記ユーザ選択手段から独立した形態で提供される。

制御される装置が選択項目セットおよびそのような選択に関する情報だけを制御する装置に提供する場合、制御する装置の便利のように、制御される装置は、それら選択項目および関連情報の表示および選択の方法を、制御する装置に委ねる。これは、いかなる制御する装置と制御される装置の間の対話に対しても本質的に同様の手法を使用することができることを意味する。制御する装置が限定的ユーザ対話能力しかもたないとすれば、非常に限られたユーザ・インターフェースが制御される装置のため構築される。しかしながら、制御する装置が豊富なユーザ対話能力を持つとすれば、そのような能力を活用して、制御する装置において制御される装置のため豊富なユーザ・インターフェースを作成することができる。

このような構成は、制御および装置動作のプロセスに対する責任を制御される装置と制御する装置の間で理想的に分割する。制御される装置の機能性およびその機能性の変更の方法は制御される装置によって理解されるだけでよい。制御する装置は制御される装置の機能性について何も理解する必要はなく、利用可能なパラメータ選択項目をどのように表示すべきかを決定するだけでよい。これによって、表示オプションを制御する装置の能力に見合うようにすることが可能とされる。

本発明は、第2の側面において、上述の方法における制御する装置の機能に対して適用される情報処理装置を提供する。該情報処理装置は、制御される装置と

の情報通信のための手段、ユーザに情報を表示してユーザ入力から導出される値を返すユーザ・インターフェース手段、および、該ユーザ・インターフェース手段を通して制御される装置から受け取った可能なパラメータ選択項目セットおよび関連情報を提示し、ユーザ・インターフェース手段からの実際のパラメータ選択項目を制御される装置へ返す手段を備える。

本発明は、第3の側面において、制御される装置における可能なパラメータ選択項目セットおよび当該可能なパラメータ選択項目を識別する情報の作成および制御する装置へのそれらの伝送を可能にする制御スクリプトを提供する。該制御スクリプトは、パラメータの各々毎に、可能なパラメータ選択項目、そのパラメータのタイプ、およびユーザ解釈可能なパラメータ情報を含む。

図面の簡単な説明

次の添付図面を参照して、本発明の特定の実施形態が例示のため以下に記述される。

図1は、本発明を適用可能な制御構成のブロック図である。

図2は、本発明の諸側面に従う制御プロセスのステップのブロック図である。

図3は、本発明の諸側面を活用することができる装置環境のブロック図である。

図4は、1つの装置に関するサーフェス印象を別の装置のサーフェス表現から作成する様態を示すブロック図である。

図5は、JetSendアーキテクチャのコンポーネントおよびそれらの論理関係を示すブロック図である。

図6(a)乃至図6(j)は、装置間の情報の交換におけるJetSend対話プロトコルのメッセージの使用を示すブロック図である。

図7は、制御する装置における仮想制御パネルの作成のためのJetSendポリシーを示すブロック図である。

図8は、仮想制御パネルを使用する制御される装置のユーザ制御のためのJetSendポリシーを示すブロック図である。

図9は、制御スクリプト符号化方式によってサポートされる最小限度のユーザ・インターフェースを示すブロック図である。

図10の(a)および図10の(b)は、制御スクリプト符号化方式から自動的に生成されるウィンドウ・ユーザ・インターフェースを示すブロック図である。

図11の(a)および図11の(b)はHTML符号化方式から生成されるユーザ・インターフェースを示すブロック図である。

図12は、一組のユーザ・インタフェース制御を定義する符号化方式を用いて生成されることができるファックス機のユーザ・インターフェースを示すブロック図である。

発明の実施形態

一般的な実施形態が図1および図2に示されている。その詳細は後述される。

図1は、本発明の諸側面の応用に関する基本的要件を示す。一定のユーザ決定可能パラメータに従って装置機能を実行する制御される装置11がある。制御される装置11は、テレビ、音楽システムまたは洗濯機のような家庭用機器、プリンタまたはスキャナのようなオフィス装置、その他の機械類または本質的に外部制御機能を持ついかなる装置でもあり得る。また、制御される装置に制御情報を提供する役割をする制御する装置12がある。制御する装置12は、また、遠隔制御装置、パーソナル・コンピュータまたは本質的にユーザ・インターフェースを持つその他のどのような装置でもあり得る。制御する装置12と制御される装置11の間には双方向通信のメカニズムすなわち2つの装置の間でメッセージを交換する手段および意味のあるメッセージが交換されるようなプロトコルの両方のメカニズムが存在しなければならない。やはり、実質的にはいかなる媒体およびプロトコルをも使用することができる。ユーザ13は制御する装置12と対話する。

本発明の1つの側面に従う制御プロセスにおけるステップが図2に示されている。制御する装置12と制御される装置11の間には(双方向に)メッセージを搬送する信号伝送手段が既に確立されていると仮定される。ユーザ13は制御する装置12のユーザ・インターフェースと対話することができる。

ステップ21は、制御される装置11による制御する装置12への可能なパラメータ選択項目のセットの提供を含む。そのような選択は、例えば、オンまたは

オフのような2者択一であることも、“1,2,3,4,5”または“'パセリ','セージ','ローズマリー','タイム'”のような一連の値からの1つの選択であったり、または、“1と12の間の1つの整数値”のような範囲からの選択である。トークンはテキスト文字列である必要はなく、適切な実施形態において、それらは例えばイメージまたはアイコンでもあり得る。本質的には、この目的のためにはどのような定義された選択項目のセットでも可能である。上記可能なパラメータ選択を識別する情報もまた提供されなければならない。例えば、“1と12の間の1つの整数値”という関連パラメータ選択に“Volume”(音量)のようなラベル、あるいは、“オンまたはオフ”という関連パラメータ選択について“Recording”(記録)というラベルが与えられる。ラベルは2重の機能を果たす。すなわち、制御される装置11がそのパラメータに対する特定の選択を反映するためその状態を変更するように適合されるように、ラベルは制御される装置11によってその機能を決定する特定のパラメータと関連づけられる。ラベルは、また、ラベル情報を備えている時ユーザ13が関連するパラメータに対する適切な選択を行うことができるようにパラメータについての十分な説明を提供する。しかしながら、ラベルは制御する装置12にとっては何の意味をも持たない。制御する装置にとってラベルは単にラベルである。制御する装置12は制御される装置11の機能に対していかなる方法でも適合される必要はない。

ステップ22は、可能なパラメータ選択項目セットおよび当該可能なパラメータ選択項目を識別する情報の表示を含む。これは、使用可能な各パラメータ選択項目を識別する情報のユーザへの提示を必要とし、ユーザがそのパラメータ選択項目に関して行うことができる可能な選択の識別と共に、所与のパラメータ選択項目に付けられるラベルのユーザへの単純な提示を含む。例えば、上のパラグラフにおいて述べたパラメータを取り上げれば、制御する装置は、“Volume”というラベルおよびそのラベルに隣接して1から12までの値を示すユーザが操作できるダイヤルまたはスライダー(滑動部)を含む表示を備えることができる。

ステップ23は、ユーザによる実際のパラメータ選択項目のセットである。これは、制御する装置12のユーザ・インターフェースにとって適切なものであればいかなる手段によっても行われる。制御される装置11によって提供されたも

の中の所与のパラメータに対する選択だけをユーザが行うことができることをこのユーザ・インタフェースが保証するという利点がある。ユーザは、制御される装置11によって提供された情報およびユーザの知識と理解に基づいて選択を行う。制御する装置12は、これらのパラメータのいずれについても選択に関する役割を持っていない。

ステップ24は、ユーザ13によって選択された実際のパラメータ選択項目の制御する装置12から制御される装置11への伝送である。制御される装置が各パラメータ選択項目がどのパラメータに関連するかを判断することができるような形態で(例えば各パラメータ選択項目への適切なラベルの添付によるなど)実際のパラメータ選択項目が与えられなければならない。

ステップ25は、実際のパラメータ選択項目に従って行われる制御される装置11の動作パラメータの修正である。

このようにして、制御する装置の機能は、制御される装置11と情報を交換することおよびユーザが制御される装置11を制御することができるようにユーザとの対話を可能にすることに限定され、これらの機能は、制御される装置11の性質および機能性に関係なく同じように動作する。制御する装置12は、ユーザに情報を表示し、ユーザがその選択を行うメカニズムの制御を持ち、従って、制御する装置12のユーザ・インターフェースの豊富さの度合いにとって適切な選択が行われる。

制御は、これらのステップを何度も繰り返す必要がある場合もある。ステップ23乃至ステップ25は、ユーザが制御される装置11の動作の変更を決定する際、何度も繰り返されることができる。(1つのパラメータ選択項目の選択が別のパラメータ選択を行うことを必要とするような)階層制御構造が存在する場合、ステップ21乃至ステップ25は、制御される装置11の装置機能の動作の前に1つのシーケンスとして1回または複数回の反復を必要とすることがある。

本発明が人間のユーザ13という観点から記述されているが、この点は、本発明の最も広範囲な側面において本発明によって必ずしも必要とされない。”ユーザ”13は、別の装置または(例えば独立したプログラムのような)別の動作面において、制御する装置12によってさえ置き換えられることもできる。ユーザ1

3について必要とされることのすべては、制御される装置11によって提示される情報に従って制御される装置11によって標示されるパラメータに関する選択を行う能力である。これは、パラメータに付加されたラベルまたはそれらラベルが含む情報を認識し、それらラベルに基づいて利用可能な選択項目から選択されたパラメータ選択項目を戻すように適合されたプログラムによって実施されることもできる。この”ユーザ”13が制御される装置11の機能の多少詳細な知識を必要とするかもしれないが、制御する装置12と制御される装置11の間の通信も、(ユーザ13がたとえ制御する装置の内部にあるプログラムであったとしても)ユーザ13への制御する装置12によるパラメータ選択項目および関連情報の提示も、制御される装置11の機能に依存しない。

人間以外のユーザの有用で実際的な例は、特定の仮想制御パネルに関して好ましいユーザ選択項目のあらかじめ定められたセットを含む制御プログラムである。制御プログラムの効果は、制御する装置12がその仮想制御パネルを”見る”時はいつでも、プログラムがそれら好ましい選択項目を戻すように動作することである。そのような制御プログラムは、既存のワード・プロセッシングまたは表計算プログラムのマクロに類似している。

上述のように、本発明は非常に一般的な応用分野を持つ。特に有用な応用分野は、制御する装置12が万能遠隔制御装置として構成されるものである。この場合の万能遠隔制御装置は、(1つまたは複数のボタン、スタイラスを含むタッチスクリーン、キーパッドその他)選択を識別できるディスプレイおよびユーザ手段、(例えば赤外線トランシーバなどの)通信手段、および受け取ったメッセージを解読し、通信手段経由の伝送のため情報をメッセージとして提供する手段を含む最適機能のユーザ・インタフェースを備える。そのような装置は、制御する装置自体は制御される装置の機能についての知識を持つ必要がないので、それと通信できるように構成された実質的にどのような制御可能な装置との対話にも役立つ装置であればよく、例えば手持ち式制御装置である。これは、遠隔制御に対する広範囲で比較的低コストの遠隔制御手法を提供し、従って、テレビ、音響装置および窓や洗濯機でさえ、それ自体はそれら制御対象装置のいずれの機能に対しても適合されていない同一の制御装置から制御することができる。

上述の動作原理に従う本発明の特定の応用分野を以下詳細に記述する。この応用分野は、ヒューレット・パッカード社のJetSendアーキテクチャの文脈に含まれている。これは、ヒューレット・パッカード社から入手可能で、WWWサイトの<http://www.jet send.hp.com/>で参照可能なHP JetSend Communications Technology Protocol Specification(JetSend通信技術プロトコル仕様)に詳細に記述されている。

JetSendは、仲介者を必要とすることなく情報を直接伝送するアーキテクチャである。このアーキテクチャの下では、情報がどのようなものに関連しようとも共通のプロトコルに従って伝送されることをネットワーク要件が可能にする。装置は、他の装置の機能性の知識を持つ必要はない。これは、コンピュータ・ネットワークの環境では、例えばスキャナおよびプリンタが、パーソナル・コンピュータの仲介なしに通信することができ、他の装置との対話を可能にするため1つの装置にロードされるべきドライバに関する必要がないことを意味する。

図3は、JetSend装置が動作する環境を図示している。(装置間の通信を使用可能にする手段の最も広い意味における)ある種のネットワーク341がプリンタ342、パーソナル・コンピュータ343およびスキャナ344のような装置の間に存在する。これらの装置の各々は、当然のことながらネットワーク341とのインターフェースを可能にする接続手段345と共に、ある種のプロセッサ346を所有しなければならない。この実施形態において、装置の各々は、JetSend全体のプロセスにとって必要ななんらかの処理容量を持つ必要がある。

JetSendアーキテクチャは、ある装置と他の装置の間の情報通信に関して効果的に利用されることができる。この点は、ヒューレット・パッカード社のその他の応用分野において、特に"Method of Passing Information"というタイトルの1997年4月15日出願の英国特許出願第9707551.9号および"HP JetSend Protocol Specification"というタイトルの1997年7月18日出願の米国特許出願第60/054,047号および"Method and Apparatus for Device Interaction by Protocol"および"Method and Apparatus for Device Interaction by Format"というタイトルの米国およびその他の国における後続の特許出願に記載されている。これら特許出願の内容は引用によって本明細書に組み込まれている。以下に示

されているように、本発明は、他の装置によるある1つの装置の特に効率的制御を達成するためJetSendアーキテクチャおよびプロトコルに容易に適用されることができる。以下、先ずJetSendアーキテクチャの基本原理およびエレメントを記述し、次に、JetSendアーキテクチャの適切な特定エレメントの文脈における本発明の適用の詳細を記述する。

JetSendにおける装置間通信に関する基礎は、サーフェス対話モデルである。サーフェス(すなわちsurface)は、装置によって所有される内部状態のなんらかの局面の表現である。表現は、普遍的であって、装置によって実行される機能によって左右されない。この文脈でのサーフェスは、(電話またはレンガ・ブロックのような)物理的実体によって提供されるサーフェスに類似している。オブジェクトが動作する形態は、オブジェクトの”機能的”部分がどのようにサーフェスに結合するかによって決定されるが、サーフェス自体は、機能にかかわらず単純な普遍的形態で記述されることが可能であり、オブジェクトがそれを經由して世の中の他の物理的実体に接続することができる媒体を提供する。従って、物理的実体の間の接続の性質はどのような装置機能からも独立している。JetSendにおいてはサーフェスは情報交換の基本単位であり、イメージ、ドキュメント、状況メッセージ、装置ラベル等々は、すべて、1つまたは複数のサーフェスの使用を通して伝送される。サーフェスは、記述、内容、特性およびクラスという多数のエレメントからなる。これらの詳細は後述される。

サーフェス対話モデルは、サーフェスを作成、共有、修正および削除するためのメカニズムを定義する。1つのサーフェスに対しては、JetSend対話プロトコル(すなわちJetSend Interaction Protocolであり以下JIPと略称される場合がある)によって定義されるメッセージを使用して固定的数の一般動作のみが実行され得る(詳細は後述)。

1つのサーフェスのオリジナルのコピーはエクスプレッション(すなわちexpression)と呼ばれる。いずれのサーフェス対話にも1つのエクスプレッションが含まれる。他の装置と共有すべき情報を持つ装置は、その情報を表すため1つまたは複数のエクスプレッションを作成する。

サーフェス・エクスプレッションは他の装置に印象づけされることができる。

これによって、そのサーフェスのインプレッション(すなわちimpression)が作成される。これはサーフェス共有としても知られている。インプレッションは他の装置のエクспレッションのコピーであり、インプレッションがその対応するエクспレッションに対して最新であることを保証するように装置間の接続が維持される。典型的には、2つの装置は、所与の時間において、ジョブのエLEMENT、各装置からの状態情報、セキュリティ情報およびその他のELEMENTを表すいくつかのサーフェスを共有する。

図4は、1つの装置(装置A)のエクспレッション1が別の装置(装置B)と共有され、それによって新しいインプレッション2が作成される様態を示している。同じエクспレッションを複数の装置に複数回インプレッション(印象づけ)することができる。これは、同じサーフェス・エクспレッションの複数インプレッションを作成する。エクспレッションが変更または削除される時、すべてのインプレッションに通知される。

サーフェスは、サーフェスの記述およびサーフェスの範囲内に含まれる内容を含む。記述は、JetSendにとって基本であり、本発明の諸側面において重要である。

サーフェス記述は、広範囲の可能な形式を確立する。この形式において、(サーフェス内容データとして提供される)サーフェス関連情報が別の装置と共有されることができる。記述は、データ形式階層から構成され、各々が(属性とも呼ばれる)データ形式の定義可能な側面を表す多数のノードを持つツリー構造として表される。サーフェスに関連する情報のための特定のデータ形式は、ルート・ノードから下降して終端ノード(すなわちツリーの葉ノード)へ至る、ツリー構造を通過する経路である。そのような特定の経路は交渉のプロセスを通して到達される。この文脈において、交渉のプロセスは、サーフェス記述またはサーフェス記述の一部を他の装置と共有するサーフェス・オーナーを含む。他の装置は望ましいオプションを選択し(一般的には、提示された選択項目を所与として利用できる最も豊富な方法でそれ自身の機能性を受信側装置が使用することができるように、そのようなオプションは選択されるが、これは、本質的に、いかなる所与の装置に関する設計者も考慮すべき事柄である)、このプロセスは、特定の経路

従って特定のデータ形式が選択されるまで、継続する。

サーフェス内容は、サーフェスに関する情報であり、上述の交渉によって決定される形式で提供される。内容データは、サーフェス記述に実施されるデータ形式階層によって示されるすべての形式で提供されることができ、所与のサーフェス対話に関して、一般的には、利用可能な形式のうちの1つでのみ提供される。内容データは交渉の完了に先だって存在する必要はない。データ形式階層における一部またはすべての選択のため、形式選択が判明した後にデータだけが生成されればよい。代替的には、一部の選択に関して、交渉プロセスの一部としてデータ形式階層における特定のポイントを識別する記述と共に内容データを直接提供する場合もある(従ってその形式選択のため内容提供の独立したステップが必要とされない)。これは、インライン内容データ提供と呼ばれる。これら諸点は、後段の制御情報の記述の際に併せて記述される。

サーフェス記述は、また、他のサーフェスに対する参照を含むことができる。そのようなサーフェスは下位サーフェス(または子サーフェス)として知られていて、それらは親サーフェスのオーナーから分離して要求されなければならない。下位サーフェスの各々は、親から独立して、それ自身の記述および内容を含む。

サーフェス参照すなわちサーフェス名は、単純に、URLのようなASCII文字列である。すべてのサーフェスはそれらの名前によって識別され、特定のクラスに属する。すべてのサーフェスは1つのクラスを所有する。クラスは、サーフェスが使用される目的を標示する特性であり、本発明の実施形態において、各サーフェスは1つのそしてただ1つのクラスを持たなければならない。特定のクラスのサーフェスの使用に関する特定の規則が存在するかまたは工夫される。クラスは特に特定のポリシーの文脈において使用される。サーフェスの特定クラスが所与のポリシーの実行における基本的役割を演ずる場合、そのクラスのサーフェスは、そのポリシーに関して“周知のサーフェス”と呼ばれる。情報を制御するように適用できるポリシーについて以下に更に記述する。

ポリシーまたは対話ポリシーは、特定のタスクを達成するため、一般的なサーフェス対話の使用に便利な規則セットである。特定のタスクのためポリシーを使用する利点は、そのポリシーによって、その特定タスクを達成するため他の装置

と成功裡にまたは予測できる形態で対話するように特定のタスクをすべてのJetSend装置が実行することが可能とされること、または、他の装置にそのようなタスクを実行するように要求することが可能とされることである。これは、例えば装置のペアがそのようなタスクを実行するため異なるサーフェス・クラスを使用する別のJetClass装置に対して開かれていない異なる方法でそのようなタスクを実行するように構成される可能性を除外しない。しかしながら、そのような装置が他のJetSend装置についてもこのタスクを達成することができるように、そのような装置が関連JetSendポリシーをやはりサポートすることが望ましい。

利用可能なサーフェス対話セットの詳細は制御情報と共に後述される。

“電子材料”(electronic material)の短縮形として使用される用語“E材料”は、サーフェスがそれを通して表現および共有される情報によってとられる形式である。それは記述および内容を含む。記述は、関連する内容データが提供されることができる形式の選択の階層を標示する。内容はサーフェスに関連するデータである。記述は、データが提示されることができる形態を標示し、一方、内容は、情報を送信する装置および受信する装置に適するように選択される形式で表される情報自体である。この選択の存在はE材料の概念にとって鍵であるので、この選択が行われるメカニズムを更に述べる。

E材料は、装置が交換するデータの形式を定義するが、装置がそのデータを処理または記憶する方法を定義しない。E材料を消費する装置は、定義した装置に特定な方法でそれを消費しない。例えば、プリンタのような受信装置は、PCのような受信装置とは異なる方法でE材料を処理する。

E材料タイプの最も基本的部分は符号化にある。符号化は、情報が一般に提示されることができる基本的形式のE材料表現である。制御アプリケーションに関して、制御対話において使用されるE材料のための特定の符号化方式が存在する。

属性は、E材料記述において定義される(または交渉プロセスの完了の後1つのE材料に関して定義される)1つのE材料の機能である。基本的には、E材料記述において提供されるデータ形式階層における各ノードは属性を表す。ノードと属性の間のこのような同等性から、“属性レベル”という概念が導出される。基

本的には、属性レベルは、データ形式階層のルート・ノードから等しい距離にあるノードのすべてによって共有される。属性は、E材料内容に対して与えられることができる品質およびその品質に関する値または選択を含み、その選択または値自体が更に別の属性(選択または値を必要する品質)を表すことができる。品質は、属性名(本明細書において単に"属性"と呼ぶ場合もある)によって表され、一方、選択または値は"属性値"(本明細書において単に"値"と呼ぶ場合もある)によって表される。従って、データ形式階層のノードは、特定の品質およびその品質に関する特定の選択またはある範囲の選択を提供するものとしてみなされることができる。属性名は、それに対して使用可能な値または選択肢があらかじめ定義されているキーワードであるので、所与のキーワードに関する所与のサーフェス記述において提供されるオプションは、これらの定義済み選択選択肢からのなんらかの選択でなければならない。異なる符号化方式に関しては、一定の属性が必要とされ、その他はオプションであるかまたは全く使用されない。

このように、E材料は、階層に構成された多数の属性値ペアでサーフェスの符号化選択項目を表す。すべてのサーフェス記述は属性符号化を含む(すべての属性および値の名前は小文字vの接頭辞を付けられる)。上記のように、これはサーフェスの総体的符号化方式を決定する。例えば、それが制御情報であれば、符号化はvControlである。各符号化は、それと関連する多数の属性一値ペアを持つ。これらの属性の一部は、また、選択項目を含むことができる。実際、サーフェスのオーナーは、サーフェス記述における属性の多くにオプションを追加することによって非常に複雑な選択項目を表すことができる。

1つの装置に関してJetSendを構成する機能性の主要な領域が3つある。それらは、装置における伝送機能、JetSendプロトコル自体および装置特有コードである。図5は、JetSendアーキテクチャのコンポーネントおよびそれらの論理関係を表している。それらコンポーネントの概要を先ず述べ、次に、各コンポーネントの詳細を記述する。図5は実施形態の図ではない点に注意する必要がある。図5は、プロトコル間の関係を示し、ソフトウェア・コンポーネントの間の関係を示すものではない。実際の実施形態は、同様のコンポーネントを持つこともあり、あるいは、複数プロトコルの実施形態を単一モジュールに結合する場合もある。

る。

JetSendアーキテクチャは、伝送から独立して適用することができる。JetSend装置は、ユニークなアドレス指定を使用して直接どのような双方向伝送機構36を経由してでも相互にアドレスすることができる。伝送機構は信頼性のあることが必要であり、従って、UDPのような信頼性の低い伝送機構の場合、その伝送機構上の伝送を信頼できるものにするため更なるプロトコル層が加えられなければならない(この目的のために、本明細書では、信頼できるメッセージ伝送プロトコル(Reliable Message Transport Protocol)略してRMTP37と呼ばれる更なるプロトコル層が使用される)。可能な伝送機構は、TCP/IP、SPX/IPX、IrDA、IEEE1284、IEEE1394およびその他を含む。装置は、同じ伝送機構を使用する他の装置と通信することを可能にするため、1つまたは複数の伝送機構を実装することができる。

JetSend装置の間の通信は、図5に見られるように、多数の層化プロトコルを使用して行われる。これらの層は、各プロトコル層が他の装置における同等の層と通信するという大部分のネットワーキング・システムに類似していて、JetSendプロトコルを含む層は、対話ポリシー35、対話プロトコル33およびセッション・プロトコル34である。RMTPプロトコル37は厳密にはJetSendの一部ではないが、信頼性および秩序に欠ける伝送プロトコルの使用を可能にするプロトコルである。

ポリシーの詳細は制御に関する記述と共に後述されるが、ポリシーは、サーフェス記述を要求および伝送し、サーフェスに関する内容データを伝送し、サーフェスを更新するメッセージを含む対話プロトコルである。E材料と関係して、ポリシーは、装置間のデータ・タイプを交渉するメカニズムを提供する。

セッション・プロトコルは、2つの装置の間でセッションおよび経路をセットするためのメッセージを定義する。セッションは、2つのJetSend実体の間のデータ伝送の確立および終了を管理する。セッションは、通信に関する文脈の範囲内の論理経路を作成することができる。下部伝送層によって提供されるいかなるものに対してもこれらの対話のすべてが発生する。また、JetSendと非JetSend装置の間のゲートウェイにおいてJSPが使用される。

チャンネルがUDPのような信頼性に欠ける伝送機構を使用する時、RMTPはそのチャンネルに関する信頼性サービスを提供する。RMTPは、信頼性のある、順序付けされた伝達プロトコルである。RMTPは、伝送機構特定ではなく、その単一インスタンスが、伝送機構スタックのすべてを経由する接続を同時に維持することができる。

装置コード(Device Code)は、JetSendプロトコルを実際の装置に結びつけるために必要とされるロジックに対して使用される用語である。図3における装置コード31は、情報を生成あるいは消費することができるどのような装置にも存在することができる。典型的生成装置はスキャナおよびPCアプリケーションである。典型的消費装置はプリンタおよびPC表示アプリケーションである。装置は、装置特定情報/データ形式とJetSendプロトコルによって使用されるE材料の間の変換を行うことができないなければならない。

伝送プロトコルに関する更に詳細の情報は、"HP JetSend Protocol Specification"というタイトルの1997年7月18日出願の米国特許出願第60/054,047号および"Method and Apparatus for Device Interaction by Protocol"および"Method and Apparatus for Device Interaction by Format"というタイトルの米国およびその他の国における後続の特許出願に記載されている。アーキテクチャおよび装置自体の動作に従ってどのような特定の装置に関しても適切な装置コードをいかに生成することができるかを当業者は理解することであろう。

JetSend対話プロトコル(JetSend Interaction Protocolであり以下略してJIPと呼称する場合がある)およびこのプロトコルを形成するメッセージについて次に記述する。また、装置の情報交換および共有を可能にするJIPによるJSPの使用についても制御情報の交換を参照しながら記述する。

JIPは、不特定多数の装置がサーフェス交換モデルに従ってサーフェスと呼ばれる情報を共有することを可能にする少数のメッセージから構成される。いかなる対話においても、1つの装置はサーフェスを所有する。オーナーのコピーはサーフェスのエクスプレッションと呼ばれ、そのオーナー自身はエクスプレッション型装置として知られる。サーフェスのその他のコピーすべてはインプレッショ

ンと呼ばれ、それらを保持する装置はインプレッション型装置と呼ばれる。JIPによって提供されるメッセージは、エクスプレッション型装置がエクスプレッションを作成および破棄し、インプレッション型装置が保持するインプレッションを破棄し、任意の装置がオリジナルのサーフェス・エクスプレッションを修正することを可能にする。

サーフェス・エクスプレッション、インプレッションその他の概念を実施するため、メッセージのリストが作成された。これらのメッセージの使用を通してすべての“サーフェス—対話”が発生する。以下のメッセージが対話プロトコルを構成する。

- ・ SurfaceMsg(インプレッション)—目標装置上にサーフェスの新しいインプレッション(impression)を作成する。また、インプレッションに対する要求を拒絶するためにも使用される。

- ・ SurfaceDeleteMsg(削除)—エクスプレッション型装置がオリジナルのエクスプレッションを削除したことをインプレッション型装置に通知する。

- ・ SurfaceReleaseMsg(インプレッション削除)—インプレッション型装置がインプレッションを削除したことをエクスプレッション型装置に通知する。

- ・ SurfaceRequestMsg(サーフェス要求)—名前をつけられたサーフェスのインプレッションを装置が要求することを可能にする。

- ・ DescriptionRequestMsg(記述要求)—当該装置がそのインプレッションを持つサーフェスに関する記述を装置が要求することを可能にする。

- ・ DescriptionReplyMsg(記述回答)—DescriptionRequestに応じてインプレッションに関する記述を伝送する。

- ・ ContentRequestMsg(内容要求)—インプレッション型装置がエクスプレッション型装置からの内容データを要求することを可能にする。

- ・ ContentReplyMsg(内容データ)—ContentRequestに応じてエクスプレッション型装置からインプレッション型装置へ内容データを伝送する。内容要求に応答する一連のこれらメッセージが存在する。このメッセージは内容要求を拒絶するためにも使用される。

- ・ SurfaceChangeMsg(変更)—情報が変更したことを装置に通知する(例えば、エ

クスプレッション型装置によるインプレッション型装置への変更通知、および、インプレッション型装置によるエクスプレッション変更の要求およびこれら要求の拒絶)。

サーフェスは多数の属性を持つ。それらは、名前、識別子、クラス、特性セット、記述、内容データおよびバージョンである。名前は、NULLで終了するASCIIストリングである。識別子は、サーフェスの各々に割り当てられて、JIPにおけるサーフェスをユニークに識別する。クラスは、サーフェスの目的を決定するために使用される。特性セットは、エクスプレッション型装置がどのJIPメッセージに応答するかを制御する。記述は、データが利用できる形式またはエクスプレッション型装置が提供する形式の記述を含む。内容データは、情報の実際のバイトを含む。バージョンは、変更メカニズムによって使用されるもので、それによって、エクスプレッション型およびインプレッション型装置がサーフェスのどのバージョンが変更に関連しているかを知ることができる。

典型的対話は次のように進行する。最初に、伝送すべき情報を持つエクスプレッション型である装置がエクスプレッションを作成する。そのため、該装置は、名前を作成し、ユニークな識別子を割り当て、特性セットを作成し、記述を作成する必要がある。サーフェス記述に記述される内容を生成することができなければならないが、この時点では、エクスプレッション型装置は内容データを作成する必要はない。

次に、エクスプレッション装置は、これらの属性を使用し、目標装置にSurfaceMsgを送ることによってこのサーフェスのインプレッションを作成することを試みる。注：そのようなSurfaceMsgsは要求なしに送られることも、別の装置から受け取られる以前のSurfaceRequestMsgに応答して送り出されることもできる。更に、SurfaceMsgを使用してインプレッションを作成するため、エクスプレッション型装置はエクスプレッションをインプレッションすべき”目標サーフェス”を持たなければならない。SurfaceMsgが以前のSurfaceRequestMsgに応答する時、この目標サーフェスの識別子は、SurfaceRequestMsgの中で検出されることができる。しかしながら、エクスプレッション型装置が要求なしにインプレッションを作成しているとすれば、目標サーフェス識別子は既存のインプレッショ

ンの識別子であり、この場合、エクスプレッションはすでに存在していなければならないか、あるいは、エクスプレッションは"デフォルトの目標"識別子に対してセットされるかもしれない。

デフォルトの目標識別子は、"作業サーフェス"と呼ばれることもある。そのようなデフォルトの存在はJIPの適切な実施形態にとって重要である。デフォルトが存在しない場合、エクスプレッション装置が最初にインプレッション装置にメッセージを送り出す時ブートストラップ問題が発生する。すなわち、エクスプレッション型装置は、(この時点で情報が無いので)インプレッション型装置のどこにインプレッションを作成すべきかわからず、エクスプレッション装置がインプレッションを作成することを望んでいることに気づかないので、インプレッション型装置は(なんらかのグローバル・メッセージを送信しない限り)エクスプレッション型装置と対話することができない。解決策は、すべての装置がデフォルトの目標識別子を持つデフォルトまたは作業サーフェスのインプレッションを受け入れることである(このような実施形態においてはデフォルト目標識別子は1にセットされる)。これは、どのエクスプレッション型装置もが目標識別子フィールドを1にセットすることによってインプレッション型装置上にインプレッションを作成することを可能にする。インプレッション型装置は、次に、(例えば、新しい目標サーフェスにインプレッションを要求するSurfaceRequestMsgメッセージによって)エクスプレッション型装置との通信を開始することができる。

JIPのメッセージの使用を図示する一連の例が図6aから図6jまでに提示されている。図6aは、本質的に図4と同じであるが、便宜上図6aとして示されている。

例1：図6a

エクスプレッション型装置が要求されていないインプレッションを作成することを望む。最初に、サーフェス・エクスプレッション121が作成される。これは、次に、SurfaceMsgを用いてインプレッション型装置にインプレッションされ、サーフェスのインプレッション122がインプレッション型装置に存在する。

例2：図6b

エクスプレッション型装置が、他の装置と交換することを望む情報に関するサー

フェス・エクспRESSIONを作成する。この例では、エクспRESSIONは要求される前に既に存在しているが、必ずしもそうである必要はない(例えば、実際に要求されるまで子サーフェスが作成される場合がしばしばある)。次に、エクспRESSION型装置が、SurfaceRequestMsgにおけるサーフェス・インプレッションに対する要求をインプレッション型装置から受け取り、それに応じてSurfaceMsgを用いてインプレッションを作成することを試みる。最終結果は例1の場合のようになり、インプレッション装置にインプレッション122が作成される。

例3：図6c

エクспRESSION型装置がサーフェス・エクспRESSIONを作成し、例1の場合と同様に、インプレッション型装置上に要求されていないインプレッションを作成することを試みる。インプレッション122は作成されるが、SurfaceReleaseMsgを用いてインプレッション型装置からエクспRESSION型装置へ即刻放出される(129)。終了状態は、サーフェス・エクспRESSION121がエクспRESSION型装置にあるが、そのサーフェスのインプレッションはインプレッション型装置にない。

例4：図6d

例1の場合のように、要求されていないインプレッション122が成功裡にインプレッション型装置にインプレッションされる。次に、エクспRESSION型装置はインプレッション122の記述を使用して、次にとるべき動作を決定することができる。この例におけるようないくつかのケースにおいては、オリジナルのSurfaceMsgに含まれるサーフェス記述は完全でない。そこで、インプレッション型装置は、Description RequestMsgメッセージを用いてエクспRESSION型装置から一層多くの情報を要求することができる。エクспRESSION型装置は、更なるエクспRESSION記述を含むDescriptionReplyMsgを用いてDescription RequestMsgに応答する。

例5：図6e

サーフェス記述が、最上位のサーフェスの下位サーフェスすなわち子サーフェスに対する参照を含むこともできる。例えば、連想として符号化されるe材料は実

際問題として常に子サーフェスを含む。例5は、子サーフェスA1を持つサーフェスAに関するものである。各サーフェスのエクスプレッション121、123がエクスプレッション型装置において提供されている(代替的にサーフェスAのエクスプレッション121だけがこの時点で提供されることもできる)。次に、サーフェスAはSurfaceMsgを用いてインプレッション型装置にインプレッションされる。インプレッション型装置は、次に、SurfaceRequestMsgを用いてエクスプレッション型装置から子サーフェスA1のインプレッションを要求することができる。この要求は拒絶されることも受け入れられることもできる。受け入れの場合、(そのようなエクスプレッションがすでに存在していないとすれば子サーフェスA1のエクスプレッションを最初に作成した後に)エクスプレッション型装置は更なるSurfaceMsgを送り出す。終了状態は、サーフェス1のエクスプレッション121および子サーフェスA1のエクスプレッション123がエクスプレッション型装置にあり、サーフェスAおよび子サーフェスA1の対応するインプレッション122、124がインプレッション型装置にある。

例6：表6f

一旦サーフェスのインプレッションがインプレッション型装置において提供されると、インプレッション型装置はContentRequestMsgを用いて内容を要求することができる。ContentRequestMsgを受け取ると、エクスプレッション型装置は、要求を拒絶するか、または、要求された形式で内容125を提供する。この内容は、(この例の場合のように)1つのContentReplyMsgメッセージ、一連のContentReplyMsgメッセージまたはストリームのような別の手段を介して送り出される。

例7：図6g

インプレッション型装置がもはやインプレッションを必要としないと判断する時(例えば、インプレッション型装置がプリンタであって、うまく印刷したドキュメントをサーフェスが表すことを確認した時)、該装置は、エクスプレッション型装置にSurfaceReleaseMsgを送り出すことによってそのインプレッションを放出することができる。この状況は、例6の状況から続く図6gに示されている。インプレッション型装置によって内容が要求され、受け取られた後、インプレッ

ションが削除されつつあることをエクスプレッション型装置に通知するためSurfaceReleaseMsgがエクスプレッション型装置へ送り戻される。そこで、エクスプレッション型装置は、インプレッションが削除されたサーフェスに関連するいかなる後続のメッセージをも無視する。

例8：図6h

エクスプレッション型装置はそれ自体エクスプレッション128を削除することができる。この実行のため、エクスプレッション型装置は、オリジナルのエクスプレッション121のインプレッション122を持つすべてのインプレッション型装置にSurfaceDeleteMsgを送る。このメッセージは、エクスプレッションが削除されたこと、および、エクスプレッション型装置が削除されたエクスプレッション・サーフェスに関連するいかなるメッセージも無視することを標示する。

例9：図6i

(エクスプレッション型装置は常に可能であるが)インプレッション型装置がエクスプレッション・サーフェスを変更することができるようにエクスプレッション・サーフェスの特性をセットすることができる。図6iは、エクスプレッション型装置によるエクスプレッション・サーフェス126の変更を示す。エクスプレッション型装置がインプレッション型装置のすべてにSurfaceChangeMsgメッセージを送信してオリジナルのエクスプレッションに対する変更があったことを標示することによってエクスプレッション・サーフェスの変更が反映される。この動作の次には、典型的には、新しい内容要求およびおそらく新しい記述要求が続く。

例10：図6j

この例では、インプレッション型装置がオリジナルのエクスプレッションに対する変更を要求する。再び、これはSurfaceChangeMsgによって行われる。これは、エクスプレッション型装置によって許容または拒絶されることができる。変更が受け入れられれば、エクスプレッション型装置は、要求元インプレッション型装置に要求を確認し、残りのインプレッション型装置に変更を通知する更なるSurfaceChangeMsgをすべてのインプレッション型装置に送る。変更が拒絶されるとすれば、エクスプレッション型装置は要求元インプレッション装置に要求が失

敗したことを通知する。

要求元インプレッション装置がエクスプレッションに対する変更を成功裡に要求した場合、該装置は、一般に、更新された内容を要求する必要はない(もっともその他のインプレッション型装置はそうする必要があるが)。なぜならば、インプレッション型装置は、一般に、自分がエクスプレッション型装置に要求した記述変更に基づいてそれ自身の内容を更新することができるからである。

JIPは、JetSendセッション・プロトコル(略してJSP)上で動く。上述のように、JSPは、セッションが使用不可能となったことの判断と共にセッションの作成および削除を含め、2つの装置の間のセッションのすべての局面を管理する。JSPは、また、基本アドレス指定、信頼性のあるメッセージ伝送プロトコルおよびJIPによって使用されるその他のどのような伝送プロトコルに対するアクセスをも提供する。上述のように、JSPは、引用によって本明細書に組み入れられている他の文献または一般に入手できるその他の文献において詳細に記述されている。また、このタイプのプロトコルに関する伝送要件は当業者によって理解されることであろう。

JetSend対話プロトコルのメッセージの各々の詳細を以下に記述する。

SurfaceMsg(インプレッション)

このメッセージは次の3つの状況において使用される：第1の状況は、エクスプレッション型装置から別の装置へのサーフェスの転送の始動である。第2に、このメッセージが別の装置からのSurfaceRequestMsgへの応答として使用される。第3に、このメッセージがエクスプレッション型からのSurfaceMsgを拒絶するために使用される。状態フィールドが、どの解釈が使用されるべきかを標示するようにセットされる。

このメッセージが、サーフェス転送を始動するため、あるいはサーフェス要求への応答として、使用される時、送信装置がそのサーフェス・テーブルにエントリを作成することによって、インプレッション型装置はどのような変更についても通知される。

メッセージを受領次第、宛先装置がインプレッションの受け入れを選択すれば、宛先装置は、そのサーフェス・テーブルにインプレッションをエクスプレッシ

ョンに関連づけるエントリを作成する。これによって、宛先装置が、サーフェスへの変更を要求する場合または変更を終了した場合をエクスプレッション型装置に通知することが可能とされる。宛先装置がインプレッションの受け入れを選択しないとすれば、宛先装置は、インプレッションを拒絶する放出メッセージを送り返すし、テーブル・エントリを作成しない。その後"インプレッション"に関する後続のいかなるメッセージも無視されなければならない。

送信装置が、インプレッションに関する放出メッセージを受け取る時、インプレッションに関するエントリをそのテーブルから削除しなければならない。これによって、インプレッションを放出した装置がそのインプレッションに関連するいかなるメッセージをも受け取らないことが保証される。

インプレッションの送信とそれを拒絶する放出メッセージの受け取りの間には短い間隔がある。この間に、エクスプレッション型装置は存在すべきインプレッションを考慮することができる。この問題は受け取り側で解決されるので、実際的な問題をなにも引き起こさない。"インプレッションづけする装置"は、受け入れられなかった、あるいは、それがもはや保持していないインプレッションに関連するメッセージを無視することを求められる。

メッセージは送信元サーフェス識別子を提供する。この識別子はエクスプレッション型装置によってサーフェスに割り当てられるもので、最初のインプレッションが作成される時点と最後のインプレッションが作成される時点の間それはユニークである。この識別子は、サーフェスをユニークに識別するためプロトコルによって使用される。0 および 1 という値が予約されている。0 はNULLのサーフェスを意味し、1 は、"デフォルト目標"サーフェス・エクスプレッションを指定するため使用される(上述のように作業サーフェスのため使用される)。

また、送信元サーフェス・クラスが提供される。これは、送信元のサーフェスのクラスであり、サーフェスの使用を決定するため使用される。各クラスの使用の詳細は、制御の記述と共に、後述される。装置がサーフェスの特定クラスを取り扱うことができない可能性がある。そのような装置は、そのようなサーフェスのすべてを無視するか、あるいは、装置が取り扱うように構成されている異なるクラスのサーフェスとしてそのクラスのサーフェスのすべてを取り扱うように構

成されている可能性がある。

本プロトコルは、使用中およびメッセージにおいて識別されるサーフェスの各々に関するバージョン番号を維持する。これは、エクスプレッション型装置がサーフェスを変更するたび毎に更新される。

インプレッションされているサーフェスの特性もまた識別される。このフィールドに関する値および関連する意味が次の表1に提示されている。

値	意味
1	エクスプレッション型装置がこのサーフェスに関する SurfaceMsgに応答する
2	エクスプレッション型装置はインプレッション型装置からの SurfaceChangeMsgを受け入れる

表1：SurfaceMsgによってインプレッションされたサーフェスの特性
これは、3という値を加えることによって他の実施形態に拡張されることができる。3という値に関して、エクスプレッション型装置はSurfaceMsgに応答して、SurfaceChangeMsgを受け入れる。

また、目標サーフェスに関するサーフェス識別子を表すフィールドが提供される。この値が1にセットされていれば、目標は、宛先のデフォルト目標サーフェスすなわち作業サーフェスであると仮定される。そうでない場合、このフィールドは、目標サーフェスの以前のインプレッションのサーフェス識別子を含まなければならない。

状態フィールドは、このSurfaceMsgの状態を識別する。次のような値が定義されている。

値	意味
0	これは要求されていないSurfaceMsgである。
1	これはSurfaceMsgに回答するインプレッションである。 要求識別子フィールドが対応する要求識別子にセットされる。
2	これは前のSurfaceRequestMsgに拒絶である。 要求識別子フィールドが対応する要求識別子にセットされる。

表2：SurfaceMsg状態に関して定義されている値

また、要求識別子フィールドがある。前のSurfaceRequestMsgの結果であるSurfaceMsgに関して、このフィールドは、対応するサーフェスSurfaceRequestMsgに含まれる要求識別子にセットされる。他のすべての状況に関して、このフィールドは0にセットされなければならない。

また、インプレッション識別子フィールドが提供される。これは、エクスプレッション型装置によって割り当てられるユニークな名前であり、同じサーフェスの異なるインプレッションを区別するために使用される。この識別子はサーフェスのインプレッションの各々に対してのみユニークであればよい。その他のサーフェスのインプレッションは同じ識別子を使用することができる。このように、エクスプレッション識別子はローカルサーフェスのすべてにわたってユニークであるのに対して、インプレッション識別子は、特定のローカルなサーフェスに関連したインプレッションのセットの範囲内でのみユニークであればよい。

更に別のフィールドは目標アドレスである。目標アドレスは、目標装置のセッション・プロトコル・アドレスである。インプレッション動作は常に目標装置および目標サーフェスを持つ。大部分のトランザクションにおいて、元および目標だけが必要とされる。しかし、本プロトコルは、1つの装置が第2の装置のインプレッションを第3のインプレッションにインプレッションすることを許容する。このケースにおいては、第3の装置は、そのJetSendセッション・プロトコル・アドレスによってのみ識別されることができる。従って、本プロトコルは、目

標装置セッション・プロトコル・アドレスの明示的指定を許容する。目標アドレスがSurfaceMsgの宛先と同じものである場合、アドレス長フィールドは0にセットされなければならない。目標アドレスは含まれてはならない。

その他のメッセージに応答するSurfaceMsgの使用が次の表3に要約されている。

応答の対象	目的	説明
メッセージなし	インプレッションの作成	非要求SurfaceMsgの送信によってエクスプレッション型装置は他の装置に対するインプレッションを作成できる。インプレッション型装置は、それを保持するか、あるいはSurfaceReleaseMsg回答でそれを放出することによって即刻拒絶することができる。
SurfaceRequestMsg	インプレッションの作成	エクスプレッション型装置は、インプレッション型装置からのSurfaceRequestMsgに応答するSurfaceMsgの送信によってインプレッションを作成することができる。
SurfaceRequestMsg	インプレッション要求の拒絶	エクスプレッション型装置は、その状態が拒絶を表すSurfaceMsgで応答することによってSurfaceRequestMsgを拒絶できる。

表3：その他のJIPメッセージに応答するSurfaceMsg

SurfaceDeleteMsgs(削除)

このメッセージは、エクスプレッションが削除されたことをインプレッション型装置に通知するためエクスプレッション型装置によって使用される。エクスプレ

レッション型装置は、エクスプレッションを削除する時、削除のインプレッションを持つすべてのインプレッション型装置に通知しなければならない。次に、エクスプレッション型装置は、そのエクスプレッションおよびすべてのインプレッションに関するエントリをそのサーフェス・テーブルから削除しなければならない。エクスプレッション型装置は、そのエクスプレッションに関連する後続のいかなるメッセージおよびそのどのようなインプレッションをも無視しなければならない。

インプレッション型装置は、削除メッセージを受け取る時、削除されたサーフェスのインプレッションに関連するすべてのエントリをそのサーフェス・テーブルから削除しなければならない。インプレッション型装置は、これらのインプレッションに関連するどのようなメッセージもそれ以降生成してはならない。

このメッセージを発するエクスプレッション型装置とそれを受け取ってそのサーフェス・テーブルからインプレッションを削除するインプレッション型装置の間に短い時間間隔がある。従って、この間に、インプレッション型装置がエクスプレッションに関連するメッセージを送り出す可能性はあるが、エクスプレッション型装置が、削除したエクスプレッションに関連するいかなるメッセージをも無視するので、実際の問題は起きない。このメッセージは、削除されるエクスプレッション識別子を必要とする。この識別子は、エクスプレッション型装置が削除しているサーフェスのサーフェス識別子である。エクスプレッション型装置は、このサーフェス識別子を含むいかなる後続のメッセージをも無視する。本プロトコルは非同期的であるので、このサーフェスを参照するメッセージがなおネットワークに存在する可能性はある。

SurfaceReleaseMsg(インプレッション削除)

このメッセージは、インプレッションが削除されたことをエクスプレッション型装置に通知するためインプレッション型装置によって使用される。インプレッション型装置は、インプレッションをもはや必要としない時、そのサーフェス・

テーブルからそれを削除して、エクスプレッション型装置にSurfaceReleaseMsgメッセージを送る。次に、インプレッション型装置は、削除されたインプレッシ

ョンに関連するすべてのメッセージを無視しなければならない。装置が同じサーフェスの複数のインプレッションを持つ可能性がある。この場合、インプレッション型装置は、削除されたインプレッションに関連する特定のメッセージのみを無視する。

エクспRESSION型装置は、そのようなメッセージを受け取る時、その特定のインプレッションに関連するエントリをそのサーフェス・テーブルから削除しなければならない。エクспRESSION型装置は、それ以降、そのインプレッションに関連するメッセージをインプレッション型装置に送り出してはならない。

このメッセージを発するインプレッション型装置とそれを受け取ってそのサーフェステ이블からエントリを削除するエクспRESSION型装置の間に短い時間間隔がある。従って、この間に、エクспRESSION型装置がそのインプレッションに関連するメッセージをインプレッション型装置に送り出す可能性はあるが、インプレッション型装置が、削除されたインプレッションに関連するいかなるメッセージをも無視するので、実際の問題は起きない。

このメッセージは、インプレッションが削除されたエクспRESSION識別子が必要とする。この識別子は、インプレッションがインプレッション型装置によって削除されたエクспRESSIONのサーフェス識別子である。これがインプレッション型装置上に残る最後のエクспRESSIONのインプレッションであれば、その装置は、このサーフェス識別子を含むそれ以降のすべてのメッセージを無視する。

また、削除されたインプレッション識別子も必要とされる。各SurfaceMsgは、エクспRESSION型装置によってユニークな名前を割り当てられる。これは、同じサーフェスの複数のインプレッションを装置が区別することを可能にする。インプレッションが削除される時、SurfaceReleaseMsgにおけるインプレッション識別子の使用によって、エクспRESSION型装置がどのインプレッションが削除されたかを判断することが可能とされる。

その他のメッセージに応答するSurfaceReleaseMsgの使用が次の表4に要約されている。

応答の対象	目的	説明
SurfaceMsg	インプレッション の放出	インプレッションがもはや不要であることをエクスプレッション型装置に通知する。
SurfaceMsg	インプレッション の拒絶	SurfaceReleaseMsgを即刻応答することによってインスプレッション型装置はインプレッションを拒絶できる。

表4 その他のメッセージに応答するSurfaceReleaseMsg

SurfaceRequestMsg(サーフェス要求)

このメッセージは、別の装置からのインプレッションを要求する装置によって使用される。メッセージは、要求側装置上にインプレッションを作成するように1つの装置に要求する。名前が、遠隔装置の有効なサーフェス名であることも、そうでないこともある。名前が有効であれば、遠隔装置は、要求側装置上にインプレッションを作成しなければならない。名前が有効でないとすれば、要求は拒絶されなければならない。目標サーフェス識別子は、遠隔装置が持つインプレッションのエクスプレッションに関する有効な識別子でなければならない。さもなければ要求は拒絶される。

エクスプレッション型装置は、サーフェス要求を受け取る時、必要に応じて要求されたサーフェスを作成し、インプレッション・メッセージを使用して要求側装置上にインプレッションを作成しなければならない。代替的には、要求が無効であれば、エクスプレッション型装置は要求を拒絶しなければならない。インプレッションまたは拒絶メッセージにおける要求識別子は、オリジナルの要求メッセージにおける要求識別子と同じものでなければならない。

要求元サーフェス名がこのメッセージのため必要とされる。これは、装置がそのインプレッションを望むサーフェスの名前である。名前は、NULLで終わる

ASCII文字列である。要求元サーフェス・クラスも必要とされる。これは、装置がそのインプレッションを望むサーフェスのクラスである。2つのエクスプレッ

ションが同じ名前を持つ可能性があり、それらはそれぞれのクラスによって区別されなければならない。要求識別子の場合のようにサーフェス要求がユニークに識別されるように、目標サーフェス識別子がまた必要とされる。

その他のJIPメッセージに関連する使用が次の表5に要約されている。

応答の対象	目的	説明
メッセージなし	インプレッションの放出	インプレッション型装置は、SurfaceRequestMsgを他の装置へ送ることによってインプレッションを受け入れることができる。このメッセージを受け取る装置は要求を応ずるあるいは拒絶するSurfaceMsgで応答する。
SurfaceMsg	子サーフェスのインプレッションの要求	インプレッションのサーフェス記述が子サーフェスへの参照を含む。インプレッション型装置は、SurfaceRequestMsgをエクスプレッション装置へ送ることによって子サーフェスのインプレッションの送信を要求することができる。

表5：その他のメッセージに応答するSurfaceRequestMsg

DescriptionRequestMsg(記述要求)

このメッセージは、当該装置がそのインプレッションを持つサーフェスに関する記述を要求するためインプレッション型装置によって使用される。インプレッション識別子は、要求側装置によって保持されている有効なインプレッションで

なければならない。エクスプレッション型装置は、更なる記述に関する要求を受け取る時、要求された記述を返すため記述回答を使用しなければならない。結果として作成される回答がデータを含まなくても、記述要求が拒絶されない可能性がある。

要求に関するインプレッションサーフェス識別子およびユニークな名前が必要とされる。

応答の対象	目的	説明
SurfaceMsg	サーフェスの記述の要求	装置がインプレッションを持つと、エクスプレッション型装置にDescriptionRequestMsgを送ることによってサーフェス記述の提供を要求する。このメッセージは、一般に、オリジナルのSurfaceMsgに含まれる記述が完全でない場合インプレッション型装置によって送られる。

表6：その他のメッセージに応答するDescriptionRequestMsg

DescriptionReplyMsg(記述回答)

このメッセージは、DescriptionRequestMsgメッセージに応答してサーフェスに関する記述を返すために使用される。無効な要求の結果は、データを含まず、これは、その特定の要求に対する記述は存在しないことを意味する。サーフェスのインプレッションを提供する装置がそのサーフェスの完全な記述を提供する準備ができていなければならないので、記述要求を”拒絶する”可能性はない。記述回答は、応答する記述要求の要求識別子を含まなければならない。

応答の対象	目的	説明
DescriptionRequestMsg	サーフェスの記述の要求	エクスプレッション型装置が、インプレッション型装置からDescriptionRequestMsgを受け取ることがある。このメッセージは、一般に、オリジナルのSurfaceMsgに含まれる記述が完全でない場合送られる。エクスプレッション型装置

は、このメッセージを受け取ると
、要求されたサーフェスの記述を
含むDescriptionReplyMsgで応答
する。

表7：その他のメッセージへの応答としてのDescriptionReplyMsg

ContentRequestMsg(内容要求)

このメッセージは、エクスプレッション型装置から内容データを要求するためインプレッション型装置によって使用される。インプレッション識別子は、要求装置によって保持される有効なインプレッションでなければならない。

要求に関する要求元サーフェス識別子およびユニークな名前が必要とされる。また、内容に関する独立した通信チャネルを確立するための情報が提供されることができる。

応答の対象	目的	説明
SurfaceMsg	サーフェスの内容 の要求	装置がインプレッションを持つ と、ContentRequestMsgの送信 によってサーフェスに含まれる 内容の提供をエクスプレッショ ン型装置に要求このメッセージ は、インプレッションを作成す るSurfaceMsgの受領後いつでも 送信できる。

表8：その他のメッセージへの応答としてのContentRequestMsg

ContentReplyMsg(内容データ)

このメッセージは、インプレッション型装置に内容データを送り出すためエクスプレッション型装置によって使用される。単一の内容要求に応答する内容データ・メッセージのシーケンスが存在することができる。このメッセージは、また、内容データに関する要求を拒絶するために使用される。

内容提供装置が1つのストリーム上にデータを提供しているとすれば、内容提

供装置は、ストリーム・フィールドをストリーム識別子にセットして、内容長およびデータを空にする。内容提供装置がストリームをサポートしないか、この目的のためにはストリームを使用することができないとすれば、内容提供装置は、ストリーム識別子を0にセットして、一連の内容応答メッセージとして内容データを伝送する。このケースでは、メッセージの内容を搬送するため内容長およびデータが使用される。

内容提供装置が内部的理由のためにもはや要求を満たすことができないとすれば、あるいは、要求が無効であれば、典型的には、内容要求を拒絶するためにだけContentReplyMsgが使用される。

すべてのケースにおいて、回答における識別子は、オリジナルの要求に含まれる要求識別子にセットされなければならない。

装置が、内容応答を受け取る時、その回答が拒絶であるか先ず判断しなければならない。もしそうであれば、装置はその次の動作を決定しなければならない。回答が拒絶でなければ、内容がストリームで供給されているか否かを装置は判断しなければならない。内容がストリームで供給されているならば、内容はそこから読み取られる。ストリームで供給されていなければ、内容は、この回答および同じ要求識別子を持つ後続の回答から読み取られなければならない(そのような回答が一連のものであるように伝送が順序づけられ、状態フィールドが一連の回答における最後を識別する)。

要求元サーフェス識別子および要求識別子が必要とされる。

状態フィールドが提供される。その適切な値が次の表9に示されている。

値	意味
0	これは内容データを含むContentReplyMsgである。この要求に関する更なう到来ブロックが存在する。
1	これは要求されたデータの最後のContentReplyMsgである。
2	これは対応する内容要求を拒絶するContentReplyMsgである。 内容データは空である。

表9：ContentReplyMsgに関する状態フィールドの有効な値

その他のJIPメッセージに応答するContentReplyMsgの使用が次の表10に提示されている。

応答の対象	目的	説明
ContentRequest Msg	要求元装置への 内容の送信	インプレッション型装置は、ContentRequestMsgの送信によって内容の提供をエクスプレッション型装置に要求する。データがメッセージ経路経由で送られるべきことをこのメッセージが示していれば、ContentReplyMsgを使用して、要求された内容が送られる。要求を満たすため複数のContentReplyMsgが送られることができる。
ContentRequest Msg	内容要求の 拒絶	2にセットされたヘッダの状態値フィールドを持つContentReplyMsgの送信によってエクスプレッション型装置は要求を拒絶する。これは、データのストリーム経路送信要求を含むすべての内容要求を

拒絶するため使用される。

表10：その他のメッセージへの応答としてのContentReplyMsg

SurfaceChangeMsg(変更)

このメッセージは、次の3つの目的のため使用される。第1に、サーフェスに対する変更をインプレッション型装置に通知するためエクスプレッション型装置によつて使用される。第2に、エクスプレッションに対する変更を要求するためインプレッション型装置によって使用される。第3に、エクスプレッションに対する変更の要求が失敗したことをインプレッション型装置に通知するため使用される。

エクスプレッション型装置は、そのエクスプレッションのうちの1つに対する変更を行う時、すべてのインプレッション型装置にその変更を通知しなければならない。これは、そのサーフェス・テーブルにおけるインプレッションを照合して、各インプレッション装置に変更メッセージを送ることによって実行される。装置が複数のインプレッションを持つ場合、各インプレッション毎に変更メッセージを送り出す必要はない。なぜなら、インプレッション装置はエクスプレッションとインプレッションの間のリンクを維持するからである。

インプレッション型装置は、変更メッセージを受け取る時、変更されたエクスプレッションのそのインプレッションの各々に対して変更を実行する必要がある。変更メッセージが変更自体を含む場合があるが、一方、メッセージが通知だけを含み、インプレッション型装置が変更されたサーフェスに関する内容を再度取り出さなければならない場合もある。エクスプレッション型装置が各インプレッション型装置によって求められている内容の符号化方式を知っていれば、エクスプレッション型装置は、関連インプレッション型装置によって求められている形式の内容を各々が含む別々の変更メッセージを提供するように構成されることができる。

インプレッション型装置は、そのインプレッションのうちの1つに対する変更を行う時、変更メッセージを使用して、エクスプレッション型装置に変更要求を送らなければならない。インプレッション型装置は、要求された変更が適用され

るエクスプレッションのバージョンをそのメッセージに含めなければならない。

変更要求メッセージを受け取り次第、エクスプレッション型装置は、変更を受け入れるべきかあるいは拒絶すべきが決定しなければならない。決定は、要求のバージョンおよび性質に基づいて行われる。エクスプレッション型装置は、適切な状態にセットされた変更メッセージを介して変更が受け入れられるがあるいは拒絶されるか要求元装置に通知する。エクスプレッション型装置は、また、前述のように、変更メッセージを使用して受け入れた変更についてすべてのインプレッション型装置に通知する。

従って、成功する変更要求を出すインプレッション型装置は、変更について2つの通知を受け取る。すなわち、1つは、(その特定の装置に送られる)変更受諾であり、もう1つは、そのエクスプレッションのインプレッションを持つすべて

の装置に送られる変更通知である。これらメッセージは、要求識別子またはバージョン情報に基づいて同じ変更に関連するものとして識別されることができる。

変更されているサーフェスのエクスプレッション識別子が必要とされる。また、オリジナルのエクスプレッションのバージョンも必要とされる。サーフェスのどのバージョンが変更されているか、また、サーフェスのインプレッションがどの状態にあるかをエクスプレッション型装置が判断することができるように、各サーフェスはバージョン番号を持つ。このフィールドは、この変更が適用される前のサーフェスのバージョン番号を含む。また、新しいサーフェス・バージョンが必要とされる。サーフェスに対する変更をインプレッション型装置に通知するためエクスプレッション型装置がSurfaceChangeMsgを発する時、このフィールドは、変更が適用された後のサーフェスの新しいバージョン番号を含む。これは、エクスプレッション型装置がバージョン番号をスキップすることを可能にする。さもなければ、このフィールドは0にセットされなければならない。

状態フィールドが提供されるが、その値は次の表11に示されている。

値	意味
0	これはサーフェスへの変更についてのエクスプレッション型装置からの通知である。
1	これはサーフェスへの変更についてのインプレッション型装置からの要求である。
2	これは前のSurfaceChangeMsgの拒絶である。変更要求は、サーフェス識別子およびオリジナル・バージョン番号によって識別される。

表11：SurfaceChangeMsgに関する状態の有効な値

その他のJIPメッセージに応答するSurfaceChangeMsgの使用が次の表12に要約されている。

応答の対象	目的	説明
メッセージなし/ SurfaceChangeMsg	エクスプレッションの変更をインプレッション型装置に通知する	エクスプレッション型装置は、そのエクスプレッションを変更すると、そのサーフェスのインプレッションを現在保有するすべてのインプレッション型装置に変更があったことをSurfaceChangeMsgの送信によって通知する。(注:エクスプレッション型装置は以前の変更要求に従ってサーフェスを変更している場合もある。)
SurfaceMsg	変更要求通知	オリジナル・エクスプレッションの変更をインプレッションの保有装置が要求できるようにインプレッションの属性をセットできる。

		インプレッション型装置はSurfaceChangeMsgの送信によって要求をエクスプレッション型装置に伝える。
ContentRequest	変更要求拒絶	エクスプレッション型装置は、受け取った変更要求の実行を望まない場合、それを拒絶して、状態値フィールドを2にセットしたSurfaceChangeMsgの送信によってインプレッション型装置に応答する。

表12：その他のメッセージへの応答としてのSurfaceChangeMsg

制御ポリシーの2つの側面に関連づけしながら、これらのメッセージの使用を制御の文脈において以下記述する。典型的には、各ポリシーは、所与のクラスの1つのサーフェスグループに関連する。従って、制御サーフェスは特定クラスを持つ(または1つのクラス・グループを与えられる)。所与のポリシーの範囲内において、所与のクラスのサーフェスは、名前またはそれらが出現する文脈によって識別される。そのようなサーフェスは、それらを交換する装置にとって既知の意味をもつので、“周知のサーフェス”と呼ばれる。ポリシーは一般にオプションである。2つの装置が1つの特定のポリシーをサポートすれば、それらは、そのポリシーを実施する情報を交換する。あるセッションに関係する1つの装置だけがある1つのポリシーをサポートする場合、そのポリシーは無視される。以下に示すように、JetSendを使用して異なるセットのメッセージを用いて制御を実行することは可能である。この制御ポリシーは、制御がどのように達成されることができるかを示す効果的な例を単に提供するものである。

ポリシーの第1の側面は、制御される装置のパラメータを決定する能力をユーザに提供する制御する装置における“制御パネル”の作成である。これは、基本的には、図2に示される一般的な制御機能のステップ21およびステップ22を提供する。使われるメッセージは図7に図示されているが、以下その詳細を記述す

る。

図7は、制御する装置12および制御される装置11を示し、また、それらの間の通信プロセスに含まれるステップを図示している。まず、ステップ91において、各装置は制御情報の交換を実行できるように初期化される。要するに、各装置は接続され、相互の存在を認識しなければならない。装置の間にセッションが確立される。

プロセスは、制御する装置12によって始動される。制御する装置12は、制御される装置11からの制御サーフェスに関してステップ92においてSurfaceRequestMsgを送り出すことによって、制御される装置11を制御する意志を標示する。正しいサーフェスが要求されていることを確認する種々のメカニズムを確立することができる。制御クラスのデフォルト・サーフェスを要求する

方式が適切なものである(代替的には、制御される装置11の適切な制御サーフェスに関する識別子が制御する装置12にすでに与えられていることである)。制御する装置12は、制御される装置11の機能の知識を必要としない。必要なことは、適切なサーフェスを確保することを確認する手段である。

制御される装置11は、この要求を受け取り、プロトコルに従って、制御する装置12にその制御サーフェスをインプレッションするため、ステップ93におけるSurfaceMsgを用いて、要求に応答する。この制御サーフェスは、ユーザがこれらの選択項目から選択を行うことを可能にするために必要な情報と共に、制御される装置11の制御のために必要とされる一連のパラメータに関する選択項目を含む。このサーフェスのインプレッション動作は実質的に図2のステップ21である。この情報は異なる符号化方式で提供されるかもしれないし、データ形式階層を通しての交渉は、DescriptionRequestMsgおよびDescriptionReplyMsgメッセージの1つまたは複数ペアを必要とするかもしれない。そのようなSurfaceMsgメッセージに関連する材料は、制御情報に関する符号化方式が取り上げられる後節において記述される。

選択されるデータ形式に関するContentRequestMsgメッセージを送出するステップ94において、制御する装置12によって符号化の選択が行われる。この後

、制御データを提供するContentReplyMsgが制御される装置12から出される。このメッセージは、なされるべきパラメータ選択のすべてを識別し、すべてのサポート情報を提供する。制御する装置12は、制御情報の作成に必要なすべての情報を持っていて、その情報は、一般に、制御パネル表示の形式でユーザに提示される。注意すべき点であるが、制御パネルは複数のサーフェス(1つまたは複数の子サーフェスを持つ親サーフェス)を含む場合もあり、その場合、このステップの完了のためには、多数のサーフェスおよび内容要求および回答が必要とされる。そのような表示を提供する情報処理は、図2のステップ22に対応するステップ96において行われる。E材料からのそのような表示の作成を更に記述する。しがしながら、表示の性質自体は制御する装置12の表示機能性に大幅に依存する可能性がある点をここで指摘することは重要である。これは、装置間で交渉される符号化において、あるいは、制御する装置が特定タイプの制御情報を表

示することができる方法において、反映されるかもしれない。これらの両方について後に更に記述する。

このようにして、制御する装置12は、制御される装置11に代わり、受け取った情報を仮想制御パネルの形式で提示する。制御する装置12は、ユーザが仮想制御パネルと対話してパラメータ値を選択することができるに十分なユーザ・インターフェースを必要とすることは明らかである。制御する装置12は、次に、これらの選択項目を制御される装置11に送り戻す必要がある。制御する装置12は、特定のパラメータの意味(例えば“電源”とラベルをつけられたパラメータが“オン”(入)に設定されることが何を意味するか)についての知識を持つ必要はないが、ユーザに許される動作のタイプについての理解を持つ必要がある。しかし、これは制御される装置11によってすでに提供されているE材料の中で提供されるべきものである。

制御ポリシーの第2の側面は、ユーザ選択を制御される装置に送り戻すステップに関連するもので、それらステップは図2のステップ23乃至ステップ25に対応する。この側面は、図8に図示されているが、更に以下に記述する。

仮想制御パネルとのユーザ対話を通して制御される装置11に送り戻されるべ

きパラメータ選択が行われた。これは、ステップ101においてSurfaceChangeMsgメッセージを用いて実行される。すなわち、制御する装置12が制御サーフェスのインプレッションだけを持つので、制御する装置は、変更を行うことはできず、制御される装置11に対して制御される装置が保持するエクスプレッションへの変更を行うことを要求することだけができる。SurfaceChangeMsgに関する上記記述において明示的に述べなかったが、このステップの後に、変更内容を提供する付加的ステップが続く場合もある。しかし、最も実地的な手法は、一般に、SurfaceChangeMsgにおいて要求されるすべての変更を識別することである。

制御される装置11は、今や、ステップ102において、更なるSurfaceChangeMsgを用いて、制御パネルに対する変更を受け入れまたは拒絶することができる。制御する装置12によって申し出された選択項目が有効で、制御される装置11によってよく理解されるものであれば、オリジナルの

SurfaceChangeMsgで識別される制御サーフェスへの変更が一般に受け入れられる。必要であれば、この時点で制御される装置によって更なる内容が提供される。仮想制御パネルは一般に現在のパラメータ選択項目を反映し、受け入れられた変更の後、更新されたサーフェスに基づくローカルの更新処理が制御する装置および制御される装置の両方において必要とされる。すなわち、制御される装置11においては、選択されたパラメータ選択項目に従って装置機能を変更することが必要であり、制御する装置12においては仮想制御パネルが採用された選択項目を反映するように変更されることが必要とされる。

次に、e材料の性質および形式を概略記述し、その後、e材料タイプおよび本発明の諸局面において有用な考慮点を記述する。

前述のように、e材料はサーフェスが表現される形式である。e材料は記述および内容を含む。記述は、内容の交換を交渉するためJetSendによって使用される。記述はサーフェスの属性を提示する。これらの属性は、典型的には、多数の属性レベルを包含する選択項目の階層を作り上げる。これらの属性は、情報がどのように運ばれることができるかを標示する。内容自体は転送されるべき知覚可能な情報である。

装置の間のサーフェスの成功裡の交換は、サーフェスを記述している各属性レベルが識別され処理されることを必要とする。装置の間でこれらのレベルを識別および処理するプロセスは交渉と呼ばれる。一旦完全なサーフェス記述が交渉されると、サーフェス内容が交換される。

2つのJetSend装置の間のサーフェスの交換は、JIPで定義されるサーフェス対話メッセージを必要とする。記述要求および記述回答を使用して、サーフェス記述は完了する。サーフェス内容交換は、内容要求および内容回答を使用して、完了する。限られた環境の下では、サーフェス内容がサーフェス記述に含まれることもある。これはインライン内容と呼ばれる。インライン内容および小さい記述を用いれば、交換はただ1つのメッセージだけを必要とするかもしれない。より一般的には、いくつかのメッセージが交換される。

E材料は、e材料ブロックの形式で提供される。1つのレベルにおいて、1つのe材料ブロックは2列のテーブルで表示されることができる。第1の列には属

性名、第2の列には属性値が含まれる。属性名は単一の特定のキーワードであり、属性値は属性に関連する1つまたは多数の値である。属性値が値のリストとして指定される場合、それらは、スペースで区切られたリストとして値フィールドに現れる。e材料ブロックに現れる属性値ペアは、JetSendサーフェスのいくつかの部分に適用できる。これらの属性値ペアは、全体のサーフェスまたはサーフェスの特定の部分に適用されることができる。すべての属性およびいくつかの値は、あらかじめ定義されたキーワードの限られたセットから引き出される。これらは、本明細書において、vEncodingのように接頭辞vによって示される。いくつかの値はデータ・タイプのセットから引き出される。これらは、本明細書において、emListTYpeのように接頭辞emによって示される。本明細書においては、JetSendサーフェスの文法的構造の考察を単純化するため定義された多数の表現タイプがある。それらは、一般的には値フィールドにおいて、適切な識別子によって標示されている。

特定の属性は、選択を申し出る値のセットに関連づけられている。値のセットの各々は選択リストと呼ばれる。選択リストは、2列テーブルにおける値の列に

スペースで区切られたリストとして現れる。交渉プロセスの間、各属性は選択リストから1つの値を取る。各値の選択は、サーフェス記述の各属性レベルの識別および処理を助ける。この実施形態における便宜性のため、リストであることができる属性データは、たとえ1つの値だけを含むとしても、リストとして符号化されなければならないという規則が設けられる。

いくつかのレベルが識別および処理されるにつれ、選択リストおよび選択された値を取り出す属性がe材料ブロックに列挙される。選択リストを取り出さない属性は除外される。このe材料ブロックは、e材料の特定の属性に関する決定を表すので、決定ブロックと呼ばれる。記述要求および内容要求は、要求されているe材料の属性を標示するe材料ブロックを含む。

属性ブロックは、サーフェス記述の特定の属性レベルに関連するすべての属性値ペアを提示するe材料ブロックである。サーフェス記述ブロックは属性ブロックの順序付けされた集合である。サーフェス記述ブロックは、3列テーブルとして表示されることができ、そのうちの最後の2つの列に属性ブロックが含まれる

。テーブルの最初の列は各属性ブロックに適用できる決定経路を含む。決定経路は、決定ブロックからの値を使用して構築される。これらは、属性ブロックと同じ順序で点記号(すなわちピリオド)で連結される。このように、例えば、イメージに適用できる属性ブロックは、`vlmage.vGray.8.(300,300).vRLE`のような決定経路によって記述される。この決定経路の例は、それが300×300dpiの解像度およびピクセル当たり8ビット・ピクセルのグレースケールのランレンジス符号化されたイメージであることを意味する。この記法は、符号化階層と呼ばれるサーフェスの合成を記述する。この階層のルートは空(null)レベルと呼ばれる。

符号化階層はツリーのような構造を形成する。この構造における各ノードは属性レベルである。あるレベルが含む属性のいずれも選択リストを取り出していないかあるいは更なる記述を必要としないとすれば、そのレベルはツリーの末端である。決定経路は、符号化階層フィールドにおいて標示されるレベルまでツリーを降下する決定の集合である。決定経路における各値は符号化階層ツリーの次の

下のレベルへの降下を示し、決定経路における最後の値は属性レベルの名前である。符号化階層におけるレベルが末端であると判明するとサーフェス記述は完了したと判断される。

サーフェス階層の範囲内のいくつかのサーフェスは1つまたは複数の子サーフェスを持つ。これらの子サーフェスは、各親サーフェス毎に明確な1つまたは複数の順序立った子リストの中に編成されている。子のリストは、1ページの片側のイメージおよびテキスト・ブロックのリストのような特徴を持つこともある。しかしながら、それは必ずしもいつでも利用できるとはかぎらない。複数ページ・スキャナによって走査されつつある多数ページのスタックを考察してみる。各ページの番号、指定および内容は存在するが、スタックの合成を反映する子リストは、任意の所与の時間において利用できる状態にはない。そのようなサーフェスは、それらが要求されるまで、あるいは、適切な装置の動作に適するなにがしか将来の時間まで、作成されない。子リストは、特徴づけられる時、親サーフェスに関連する子サーフェスの名前を与える。サーフェス記述テーブルにおいて、これは参照リストと呼ばれる。

決定経路は、決定ブロックにおける値に基づいて構築される。これらの値は選択リストから取り出される。符号化階層は、これらの決定経路の観点から表現される。決定経路は、各属性ブロックのレベルを識別するために使用される。

e 材料の文法および用語の詳細は、“HP JetSend Protocol Specification”というタイトルの1997年7月18日出願の米国特許出願第60/054,047号および“Method and Apparatus for Device Interaction by Protocol”および“Method and Apparatus for Device Interaction by Format”というタイトルの米国およびその他の国における後続の特許出願、およびWWWウェブ・サイト<http://www.jetsend.hp.com/>のようなその他一般に利用可能な資料に記載されている。

すべての環境の下でJetSend装置が通信することを保証するためには、一般的には特定の決定経路が存在しなければならない。これらはデフォルト符号化と呼ばれる。送信側装置は、このようなデフォルト決定経路によって記述される属性を用いて e 材料を生成することができなければならない。同様に、受け取り側装

置は、これらの特徴を持つ e 材料を解釈することができなければならない。デフォルトの符号化は、装置が通常状態の下で e 材料を交換することを保証するために存在する。これらの符号化は、最も低い共通特徴である(注意すべきは、この点は既存のサーフェスの変更には必ずしもあてはまるというわけではない。すなわち、例えば上述のような制御ポリシーにおける制御する装置 1 2 のようなインプレッションを持つサーフェスが SurfaceChangeMsg メッセージを通して変更を要求するとすれば、そのインプレッションに関してすでに選択されたもの以外のどのような符号化にもそのメッセージが関連するように求められることは実際的ではない)。基本符号化は、装置が一層高い忠実性で e 材料を交換することを可能にする推奨決定経路である。その他の属性および値はオプションとみなされる。

デフォルト符号化は、そのエレメントの各々が選択リストに基づく複数決定経路である。属性は各選択リストに関連づけられる。その値がデフォルト符号化を形成する属性の各々はサーフェス記述に存在しなければならない。

これら原理の制御への適用を以下記述する。最初に、制御符号化の必需機能および任意機能を考察する。

制御符号化が役立つためには、それが 2 つの基本要件に対応することができる必要がある。これらの要件のうちの 1 つは、元々発した装置以外の装置上で提示されることができるとような方法で符号化が制御パネルを表すことができなければならないことである。もう 1 つの要件は、変更の表現を受け取る装置が変更の表現を曖昧な点のないように解釈することができるとような形態で制御パネルに対する変更を表現する方法を符号化が定義しなければならないことである。

その他の機能は必須というよりも望ましいものである。1 つは、符号化がプラットフォームの範囲で利用可能であるという点である。任意の可能な制御する装置によって理解されることができると少なくとも 1 つの符号化が提供されるということとを前提とすれば、代替的符号化が提供され得るので、この点は必須ではない。視覚的制御パネルが可能な限り多くの異なる装置上で表示および処理されることを可能にするのが理想的な符号化であるが、ユーザ・インターフェース設計者がユーザ・インターフェースの様相および動作を正確に指定することができる能力

と仮想制御パネルが広範囲にわたる機能を備えながら装置上に提示されることを可能にすることができるほどの抽象化達成度の間に妥協点が存在する。しかしながら、高度に特定の符号化でさえ、装置オペレーティング・システムのような装置タイプの機構ではなく、(画面サイズや入力装置のような)装置の機能が、符号化が受け入れられるか否かを決定すべきである。高度の抽象化は、異なる能力の装置に関する広範囲にわたる表現の可能性を許容する。この点は、制御に関する必須の符号化の選択(ControlScript)に関連して更に後述される。

更に別の望ましい機能は、仮想制御パネルおよびそれになされたいかなる変更をも表示するため符号化がデータをほとんど必要としない点である。いかなる通信チャンネルのバンド幅も有限であり、JetSendのネットワーク独立性がいかなる通信チャンネル上でもJetSendを利用することができることを意味するので、仮想パネルの確立および使用の際の非常に長い応答時間の可能性を最小にするため、この機能は望ましい。

符号化に関する交渉が発生することは、JetSendにおける制御の動作に関して必須のものではない。標準的制御スクリプト符号化の下でおよびただ1つの制御スクリプトが提供されるような形態で、すべての制御動作が指定されることは可

能である。しかしながら、異なる装置に最適の表示を行うことができるように複数の符号化が提供されることは望ましい。本明細書における代替的符号化の例はHTMLである。HTML符号化のエレメントの概要は、ControlScriptについての記述の後に、記述される。有効なもう1つの符号化の選択肢は、Javaアプリケーションを使用して、制御選択項目およびそれらの表現を記述することである。所与の符号化に関する交渉が発生することもまた望ましいことが多い。例えば、ControlScriptが要求されるとすれば、交渉を通して確立されるものであるが、制御する装置12の能力に従って異なる制御スクリプト内容が提供されることもできる。

次に、ControlScriptを記述する。ControlScript(すなわち制御スクリプト)は、制御パネルのサーフェスを定義することを試みるというよりはむしろ一連の制御オプションを記述するように設計されている。制御する装置の表示または入力

能力に関する最小限度の数の仮定が作られる。従って、非常に単純な制御する装置でさえ、複雑で洗練された制御される装置に対して使用することができる。しかしながら、洗練された対話ができる制御する装置は、それらの能力に適するユーザ・インターフェースを提供するようにControlScript記述を使用することができる。このように、制御スクリプトは、最小限度の制御する装置からデータ・グローブまたは仮想現実ヘッドセットに至るまでのいかなる制御する装置に関する符号化として使用されることができる。

ControlScriptは、必須の符号化としての使用を意図されているので、最小限度のユーザ・インターフェースの使用にも適用される。制御スクリプトが最小限度のユーザ・インターフェースの一定の機能を必要とすることは明らかに望ましい。しかし、例えば1つのLEDおよび1つのボタンによる制御に適切な符号化を必須の符号化として設計することは、そのような符号化が大部分の制御パネルに全く効果を発揮しないので、実際的ではない(原理的にはそのような符号化は本発明に従って設計し、使用することができるが、それは特定の限られた環境において役立つだけである)。ControlScriptのために選択される制限は、6つの英数字からなる2行を4つの制御ボタンと共に示すことができる表示である。そのような表示のコンピュータ・シミュレーションが図9に提供されている。

図9の最小限度の表示201は、第1の表示行206および第2の表示行207を持つ。第1の表示行206の左および右に第1のボタン202および第2のボタン203がある。第2の表示行207の左および右に第3のボタン204および第4のボタン205がある。そのような最小限度の表示の提案される使用方は、第1の表示行206にパラメータ・ラベルを表示して、そのパラメータに関する選択項目を第2の表示行207に表示するものである。第1および第2のボタン202、203は、選択のため使用できるいくつかのパラメータを前後に移動させるために使用される。第3および第4のボタン204、205は、第1の表示行に標示されるパラメータに関して使用できるいくつかの選択項目を前後に移動させるために使用される。

ControlScript(制御スクリプト)に関する典型的な文法を以下に提示する。パ

ラメータが1つまたは2つのタイプの値だけをとることができることは、上述の最小限の装置ユーザ・インターフェースの記述から明らかである。それらのタイプは、可能性のリスト(列挙された選択項目)から選択される文字列、または、上限と下限(ある範囲)の間に設定される数である。文法はここに明示的に定義されないが、イタリックで示される制御スクリプトでこれらのタイプの各々に関する例を以下に示す。

列挙される選択項目：Channel:[BBC1/BBC2/ITV/C4]

このスクリプトは、ユーザが多数のオプションから選択することができることを示す。このケースにおいて、“Channel”(チャンネル)というラベルをつけられたパラメータは、“BBC1”、“BBC2”、“ITV”または“C4”という値を持つことができる。記号“[”および“]”は利用可能な値の開始および終了を表し、“/”は論理的排他的ORを表す。

範囲：Copies:[1...10]

この例では、“Copies”(コピー枚数)というラベルを付けられたパラメータは、ユーザに1と10の間の任意の値を選択させるものである。この値が整数値であるように文法が定義されるであろう。

混合タイプ：Volume:[Mute/1..10]

範囲は列挙された選択項目リストにおけるエレメントであるかもしれない。この

例では、“Volume”(音量)は値“Mute”(音消)または1と10の間の数をとることができる。

選択項目の結合

```
Television[
    Power:[on/off]
    & Volume:[Mute/1..10]
    & Channel:[BBC1/BBC2/ITV/C4]
]
```

このスクリプトにおいては、論理的AND演算を標示する“&”記号を使用して1つのグループを形成するように多数の制御が結合されている。外の大かっこは、メタ

制御Television(テレビ)を定義するため使用されている。(この例において仮想制御パネルに対するラベルとしての役目をする)メタ制御は、“on”(オン)および“off”(オフ)という値をとることができる“Power”(電源)制御および“Volume”と“Channel”の制御から構成される。

複合制御

```
Camera[
  Manual:[
    Aperture:[1..50]
    & ShutterSpeed:[1..50]
    /Pre-set:[Pre-set1/Pre-set2/Pre-set3]
    /Automatic
  ]
```

単一のカメラに関するこの複合制御は、上述された諸機能を結合することによって得られる。ユーザは、3つの異なるモードの任意の1つにカメラをセットするように選択することができる。“Manual”(手動)において、ユーザはAperture(絞り)とShutterSpeed(シャッター速度)の両方に関する値を指定しなければならないが、一方、“Pre-set”(プリセット)においては、ユーザは3つの事前に設定されたモードから選択しなければならない。“Automatic”(自動)においてはユーザはそれ以上の入力を必要としない。

最小限度のユーザ・インターフェースへのこれらスクリプトの適用は簡単明瞭である。例えば上述のカメラの場合には、デフォルト選択項目に関して発生するパラメータだけが、ボタンを用いてスクロール可能なオプションとして最初に提示されることができ、その他のパラメータはカメラのモードに関して異なる選択が行われるときに現れる。(ボタン、プルダウンメニュー、スライダー、ラジオ・ボタン、ディレクトリ・ツリー等々を持つ)一層豊富に指定されるユーザ・インターフェースの場合、適切なユーザ・インターフェースが生成されることを可能にするための規則セットが設けられることが必要である。表示装置自体が(実際の情報内容に関係なく)異なるクラスの情報を表示する方法だけに規則は関連

するので、そのような規則のセットは装置特定(またはユーザ・インターフェース特定)であり得る。共通のユーザ・インターフェースのためのスタイル・ガイドが利用できるので、規則を追加することによって、自動的ユーザ・インターフェース生成のためのプロセスにそれらが容易に変換可能であることは当業者に認められることであろう。ユーザ・インターフェース生成の例は次のようないくつかの文献によって提供されている。

D. Olsen著”A Programming Language Basis for User Interface Management”Human Factors in Computing Systems CHI89 Conference Proceedings May 1989 pp 171-176(組み込みPascalからMacintoshユーザ・インターフェースの生成に関するもの);

C. Wiecha他著”Generating Highly Interactive User Interfaces”, Human Factors in Computing Systems, CHI’89 Conference Proceedings, May 1989, pp277-282(Motifダイアログ・ボックス生成に関するもの);

B. Vander ZandenおよびB. A. Myers著”Automatic, Look-and-feel independent Dialog Creation for Graphical User Interfaces”, Human Factors in Computing Systems, CHI’90 Conference Proceedings, April 1990, pp27-34(Motifダイアログ・ボックス生成に関するもの);

D. deBaar他著”Coupling Application Design and User Interface Design”, Human Factors in Computing Systems, CHI’92 Conference Proceedings May 1992, pp 259-266(OpenLook GUIに関するもの);

J. Johnson著”Selectors Going Beyond User Interface Widgets”Human Factors in Computing Systems, CHI’92 Conference Proceedings, May 1992, pp273-279(サーフェスではなくむしろインターフェース意味論に従って分類されるオブジェクトからユーザ・インターフェースの半自動式構築に関するもの)。

比較的豊富なユーザ・インターフェースのための制御スクリプトから仮想制御パネルを自動生成する例として、次の規則セットに従ってダイアログボックスを生成する擬似コード・プログラムが以下に提供される。

制御選択項目

ControlScriptがユーザ・インターフェース制御に関する情報を含まないので、プログラマは、対象ユーザ・インターフェースに関する望ましい選択項目を自由に作成することができる。

1. 範囲のみによって記述されるいかなるパラメータも、パラメータ名でラベルをつけられるスライダー・バーによって表される。
2. 5つまたはそれ以上の単純な選択項目(例えばそれ以上の制御がセットされることを必要としない選択項目)を持ついかなるパラメータもリスト・ボックスによって表される。
3. 範囲の選択または一つの代替的選択項目によって記述されるパラメータはスライダー・バーおよびチェック・ボックスによって表される。
4. 残りのいかなるパラメータもラジオ・ボタンの集合によって表される。

制御項目のレイアウト

1. 制御項目は、ControlScriptに出現する順序でダイアロ・グボックスの上から下へ置かれる。
2. その使用がセットされつつある他の制御項目に依存する制御(下位制御)は、“親”制御項目に対してインデントされる。
3. ダイアロ・グボックスは、その範囲内に制御を収納することができるようにサイズを変えられる。

擬似コードは、制御スクリプトを解析して代替データ構造を作成し、次に、こ

のデータ構造および上述の基本的規則のセットから、各制御エレメントをどの部品が表すべきかを計算する。次に、疑似コードは、関数セット(本明細書に記述されていない)を呼び出して、これら部品を生成して上記規則に従ってダイアログ・ボックス上に配置する。疑似コードは、特定のオペレーティング・システムに固有のものではないが、この疑似コードを(マイクロソフト・ウインドウズのような)所与のオペレーティング・システムに適用する方法を当業者は容易に理解することであろう。

```

/* ControlElementType - structure used represent each element in the Control Script */
struct {
    String          Name;
    ControlElementType *NextSibling;
    ControlElementType *PrevSibling;
    SelectorType     *ParentSelector;
    SelectorType     *ChildSelector;
    ControlWidgetType Widget;
} ControlElementType;

/* SelectorType - structure used to link a Control Element to the various values it can take */
struct {
    SelectorType     *NextSelector;
    SelectorType     *PrevSelector;
    ControlElementType *ParentControlElement;
    ControlElementType *ParentControlElement;
} SelectorType;

/* ControlWidgetType - union of all available types of control elements (widgets, etc) */
union {
    "GROUP_OF_RADIO_BUTTONS"
    "SLIDER_BAR"
    "CHOICE"
    .
    .
    .
} ControlWidgetType;

/** MAIN FUNCTION **/
Main()
{
    ControlElementType *TheControlStructure;

    TheControlStructure = ParseControlFromString(ControlScript, NULL);
    AnalyseControl(TheControlStructure);
    CreateControlWidget(TheControlStructure);
    ResizeDialogBoxToFitControls(...);
}

/** PARSE THE CONTROL SCRIPT INTO A DIFFERENT DATA STRUCTURE **/
ControlElementType* ParseControlFromString(String *ControlScript, void *ParentSelector)
{
    // Read the next Symbol from the control script
    String Symbol(GetNextSymbol(TheString));

    // Initialise the record of this Control Element
    p_ControlElement = new ControlElementType;
    p_ControlElement->Name, Symbol);

```

```

p_ControlElement->NextSibling = NULL;
p_ControlElement->PrevSibling = NULL;
p_ControlElement->ChildSelector = NULL;
p_ControlElement->ParentSelector = ParentSelector;

p_FirstControl = p_Control;

// Read the next Symbol from the control script
String Symbol(GetNextSymbol(TheString));

while ( (Symbol != "]" ) && (StringLength(TheControlScript)>0) ) {
    // While not the end of the Control Element or the end of the Control Script

    if (Symbol == "[") { // Create a new Selector to be the child of this Control
Element
        p_NewSelector = new SelectorType;
        p_ControlElement->ChildSelector = p_NewSelector;
        p_NewSelector->NextSelector = NULL;
        p_NewSelector->PrevSelector = NULL;
        p_NewSelector->ParentControlElement = p_ControlElement;
        p_NewSelector->ChildControlElement =
ParseControlFromString(TheControlScript, p_NewSelector);
    }
    else if (Symbol == "[") { // Create a new Control Element to be a sibling to
this Control Element
        p_NewControlElement = new ControlElementType;
        p_ControlElement->NextSibling = p_NewControlElement;
        p_NewControlElement->PrevSibling = p_ControlElement;
        p_NewControlElement->NextSibling = NULL;
        p_NewControlElement->ChildSelector = NULL;
        p_NewControlElement->ParentSelector = ParentSelector;
        p_NewControlElement->Name = GetNextSymbol(TheString));
        p_ControlElement = p_NewControlElement;
    }
    else if (Symbol == "&") { // Create a new Selector to be a sibling to the
current Selector
        p_NewSelector = new SelectorType;
        p_CurrentSelector = p_ControlElement->ParentSelector;
        p_ParentControlElement = p_CurrentSelector->
>ParentControlElement;
        p_CurrentSelector->NextSelector = p_NewSelector;
        p_NewSelector->PrevSelector = p_CurrentSelector;
        p_NewSelector->NextSelector = NULL;
        p_NewSelector->ParentControlElement =
p_ParentControlElement;
        p_NewSelector->ChildControlElement =
ParseControlFromString(TheControlScript, p_NewSelector);
    }

    // Read the next Symbol from the control script
    String Symbol(GetNextSymbol(TheString));
}
return p_FirstControl;
}

/** DETERMINE WHICH WIDGET TYPE TO USE FOR EACH CONTROL ELEMENT */
ControlWidgetType AnalyseControl(ControlElementType *ControlElement)
{
    TheSelector = ControlElement->ChildSelector;

```

```

if (TheSelector != NULL) {
    // if the Control Element has at least one Child Selector;

    NumberOfChoices = 0;
    NumberOfRanges = 0;
    NumberOfSubControls = 0;

    do {
        SubControl = TheSelector->ChildControl;
        do {
            ControlType = AnalyseControl(SubControl);
            if (ControlType = "RANGE") {
                NumberOfRanges++;
            }
            if (ControlType = "CHOICE") {
                NumberOfChoices++;
            }
            if (ControlType, "SUB_CONTROL") {
                NumberOfSubControls++;
            }
        } while( (SubControl = SubControl->NextSibling) !=NULL );
    } while( (Selector = Selector->NextSelector) !=NULL );

    if ((Ranges==1)&&(Choices==0)) {
        // This control element will be represented by a slider bar;
        ControlElement->Widget = "SLIDER_BAR";
    }
    if ((Ranges==1)&&(Choices==1)) {
        // This control element will be represented by a slider bar and a
        check box;
        ControlElement->Widget = "SLIDER_BAR_AND_CHECK_BOX";
    }
    if ((Choices<5)&&(Choices>1)) {
        // This control element will be represented by a group of radio
        buttons;
        ControlElement->Widget = "GROUP_OF_RADIO_BUTTONS";
    }
    if ((Choices>=5)) {
        // This control element will be represented by a list box;
        ControlElement->Widget = "LIST_BOX";
    }
    if (Controls>0) {
        // This control element will be represented by a label only;
        ControlElement->Widget = "LABEL";
    }
    return "SUB_CONTROL";
}
else {
    int Min, Max, Value;
    if (sscanf(ControlElement->Name, "%d..%d..%d", &Min, &Max,
    &Value)==3) {
        ControlElement->Widget = "RANGE";
        return "RANGE";
    }
    else {
        ControlElement->Widget = "LABEL";
        return "CHOICE";
    }
}
}

```

```

}

/** CREATE THE WIDGET FOR EACH CONTROL ELEMENT */
void CreateControlWidget(ControlElementType* ControlElement)
{
    DistanceFromTop = 10;
    IndentFromLeft = 10;

    if (ControlElement->Widget == "LABEL") {
        if (ControlElement->ParentSelector != NULL) {
            if ( (ControlElement->NextSibling==NULL) && (ControlElement-
>PrevSibling==NULL) ) CreateLabel(ControlElement->Name, ...);
            else CreateRadioButton(ControlElement->Name, ...);
            PreviousIndentFromLeft = IndentFromLeft;
            IndentFromLeft += 20;
            DistanceFromTop += 20;
        }

        Selector = ControlElement->ChildSelector;

        do {
            NewControlElement = Selector->ChildControlElement;
            do {
                CreateControl(NewControlElement);
            } while( (NewControlElement = NewControlElement->NextSibling)
!= NULL );
        }while( (Selector = Selector->NextSelector) != NULL );

        if (ControlElement->ParentSelector != NULL) {
            IndentFromLeft = PreviousIndentFromLeft;
            DistanceFromTop += 10;
        }
    }

    else if (ControlElement->Widget == "GROUP_OF_RADIO_BUTTONS") {
        if ( (ControlElement->NextSibling == NULL) && (ControlElement-
>PrevSibling == NULL) ) CreateLabel(ControlElement->Name, ...);
        else CreateRadioButton(ControlElement->Name,...);
        PreviousIndentFromLeft = IndentFromLeft;
        IndentFromLeft += 20;
        DistanceFromTop += 16;

        Selector = (SelectorType*)ControlElement->ChildSelector;
        do {
            NewControl = Selector->ChildControl;
            do {
                CreateControl(NewControl);
            } while((NewControl = NewControl->NextSibling) != NULL);
        } while( (Selector = Selector->NextSelector) != NULL);
        IndentFromLeft += PreviousIndentFromLeft;
        DistanceFromTop += 10;
    }

    else if (ControlElement->Widget == "SLIDER_BAR" ) {
        if ((Control->NextSibling==NULL)&&(Control->PrevSibling==NULL))
CreateLabel(ControlElement->Name, ...);
        else CreateRadioButton(ControlElement->Name, ...);
    }
}

```



```

        CreateSliderBar();
    }
    else if .....
    }
    .
    .
    .
}

```

上述の規則は当然のことながら典型的な例である。上記およびその他のユーザ・インターフェースに関して、利用可能なユーザ・インターフェース能力を適切に活用しながら、そのような望ましい規則のいかなる選択も可能である。

“Television”および“Camera”という上記の例に関する仮想制御パネルが図10(a)および図10(b)にそれぞれ示されている。図から見ることにできるように、ウィンドウ・ユーザ・インターフェースの完全に満足できる使用が行われている。しかしながら、そのようなユーザ・インターフェースに関して、ユーザ・インターフェースの機能性の一層豊富な使用を可能にする一層強力な符号化を交渉することが望ましい場合もある。これは、なお制御スクリプトと呼ばれるが、基本文法に入れられてないオプションのエLEMENTを含む。例えば、特定のスクリプトは、カラー・ホイールまたはレンダリング・パターンまたはその他のイメージのため特定のスクリプトが提供されることができる。

代替的手法は、レイアウト情報と共に装置の目標範囲によって使用されることができる制御セットを定義する符号化を提供することである。実施可能な制御部品は、例えば、“push”(押す)によって変更可能で起動時に値を返す電話機型押しボタン(ディスプレイ・キーパッド作成の際に役立つ)、“push”によって変更可能で起動時に2つの選択項目から1つの値を切り換え選択する2状態ボタン(オン/オフ・スイッチ)、戻すことができる値を提供する新しいスライダー位置を持つ変更可能なスライダー、メニュー項目の選択によって変更可能な値を持つメニュー、および新しいテキストのユーザ入力に応じて変更可能な値を持つテキストである。そのような制御部品および配置情報からユーザ・インターフェースを作成することができる。図12は、ファックスのユーザ・インターフェースを示し

ている。これは、電話機型押しボタン301、2状態ボタン302、メニュー304およびテキスト・フィールド305を含む。適切な実施形態において、別個のキーボード入力によってまたは電話機型ボタン301で作成される仮想キーボードの使用によって数がテキスト・フィールド305に入力される。

代替的には、異なる符号化タイプが交渉されることもできる。1つの例は、Forms拡張を使用するHTML符号化である。HTML符号化において任意のテキスト文字列、基本的レイアウト情報および部品タイプが定義されるが、結果として生成されるページのサーフェスは大部分クライアントによって決定されることができるので、広範囲にわたる表示に適合されることができる。

上述の例である"Power"に関するHTMLスクリプトの例の一部は次の通りである

。

<HR>

<H1>PowerControl</H1>

Click to turn the powerON

orOFF

<HR>

これはForm拡張のない場合である。

<HR>

<H1>power</H1>

<INPUTTYPE=SUBMITVALUE="On">

<INPUTTYPE=SUBMITVALUE="Off">

<HR>

これは、図11の(a)および(b)にそれぞれ示される制御する装置によって提示されることができる。

本明細書に記述される1つの装置を他の装置から制御する方法は、JetSendプロトコルの文脈において価値あるものとしてまた効果的に実施される。しかしながら、上記に示されているように、本発明は、1つの装置の他の装置による制御における非常に一般的応用を有していて、JetSendおよび同等のプロトコルにお

ける応用に制限されるものではなく、1つの装置の他の装置による制御のほとんどいかなる問題にも適用できるものである。

【図1】

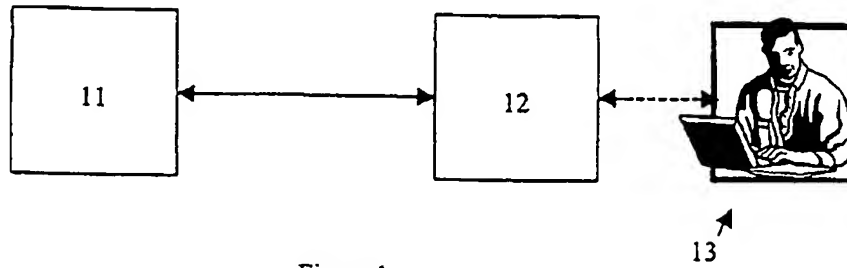


Figure 1

【図2】

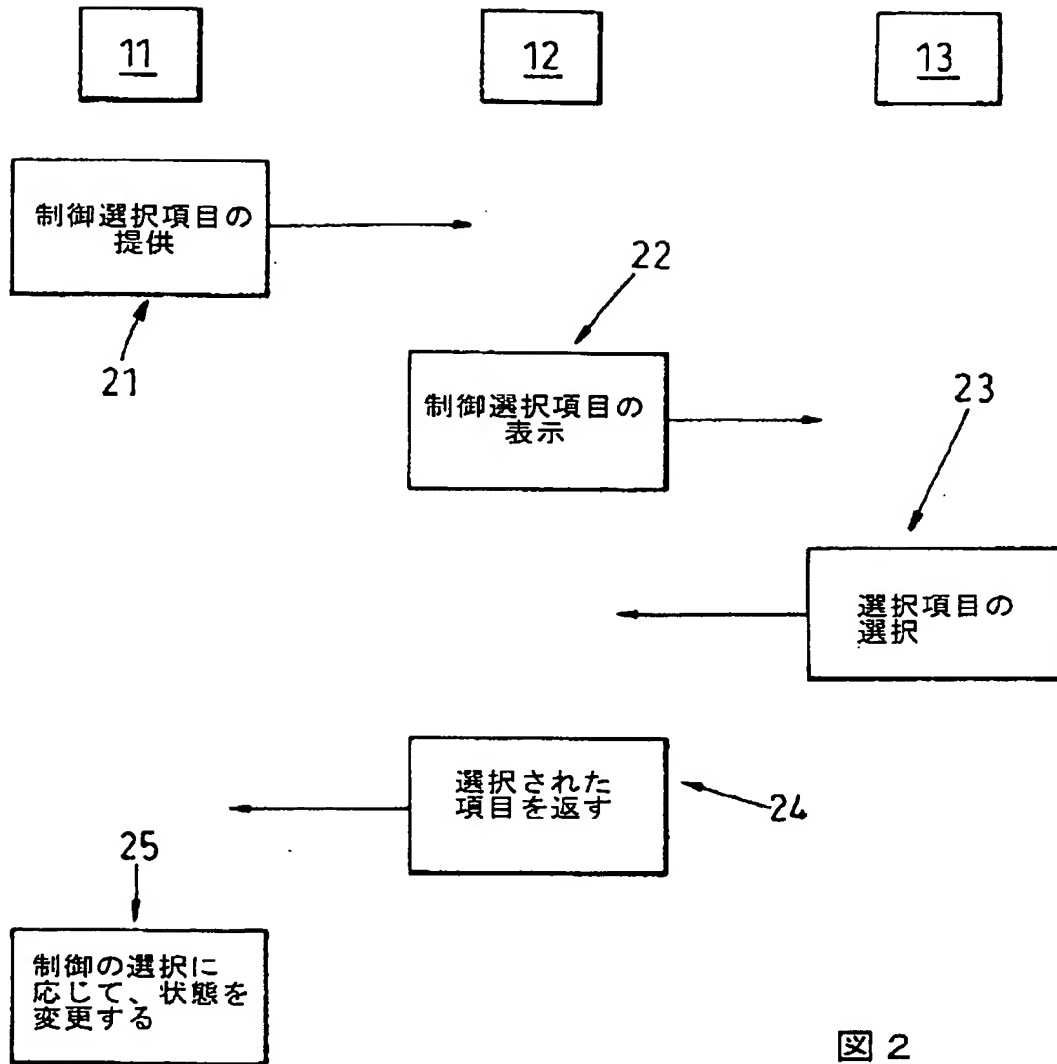


図 2

【図3】

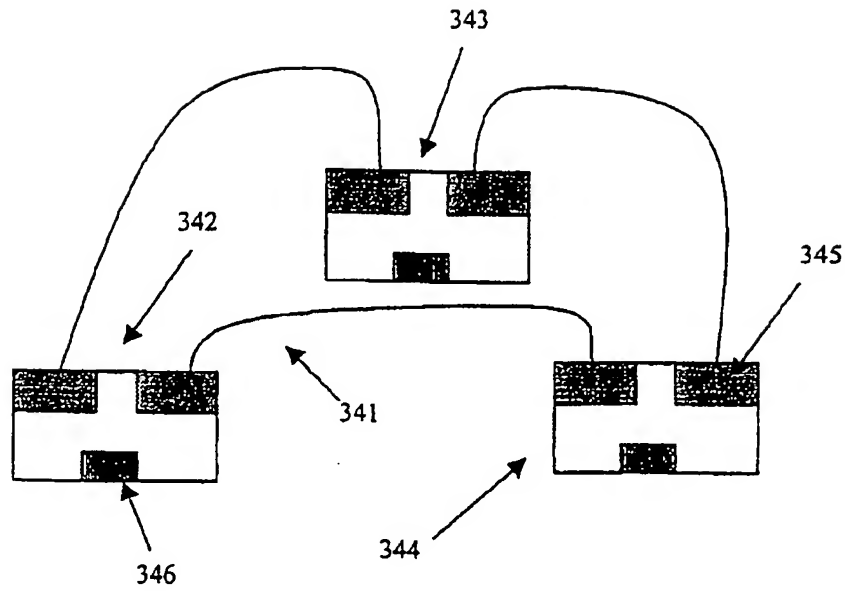


Figure 3

【図4】

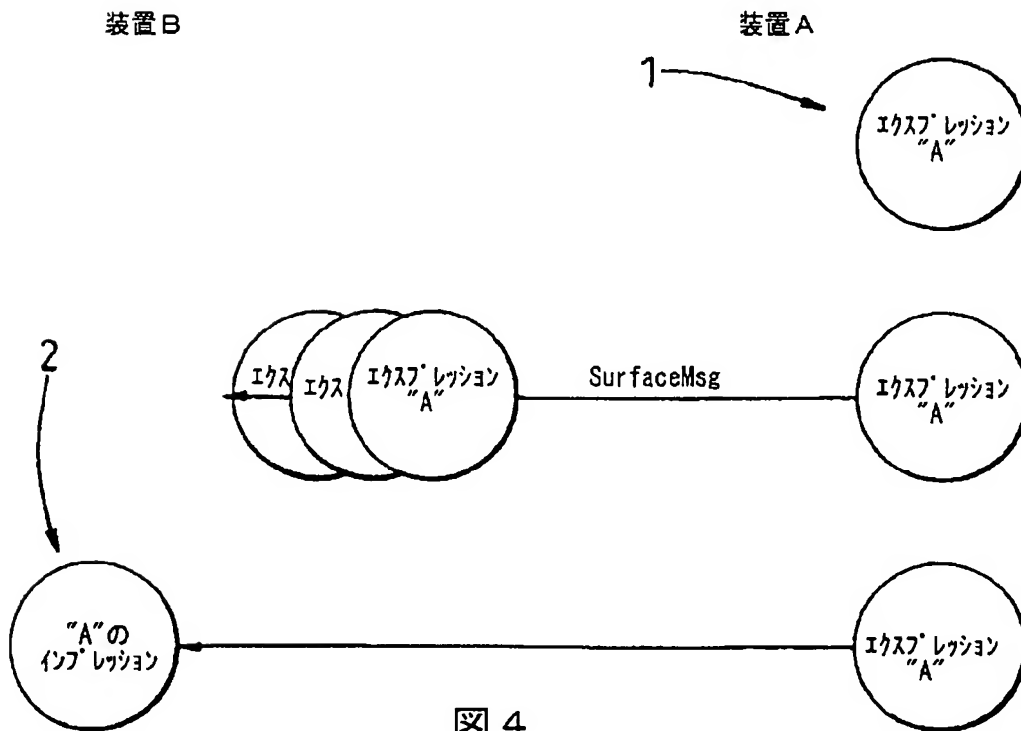
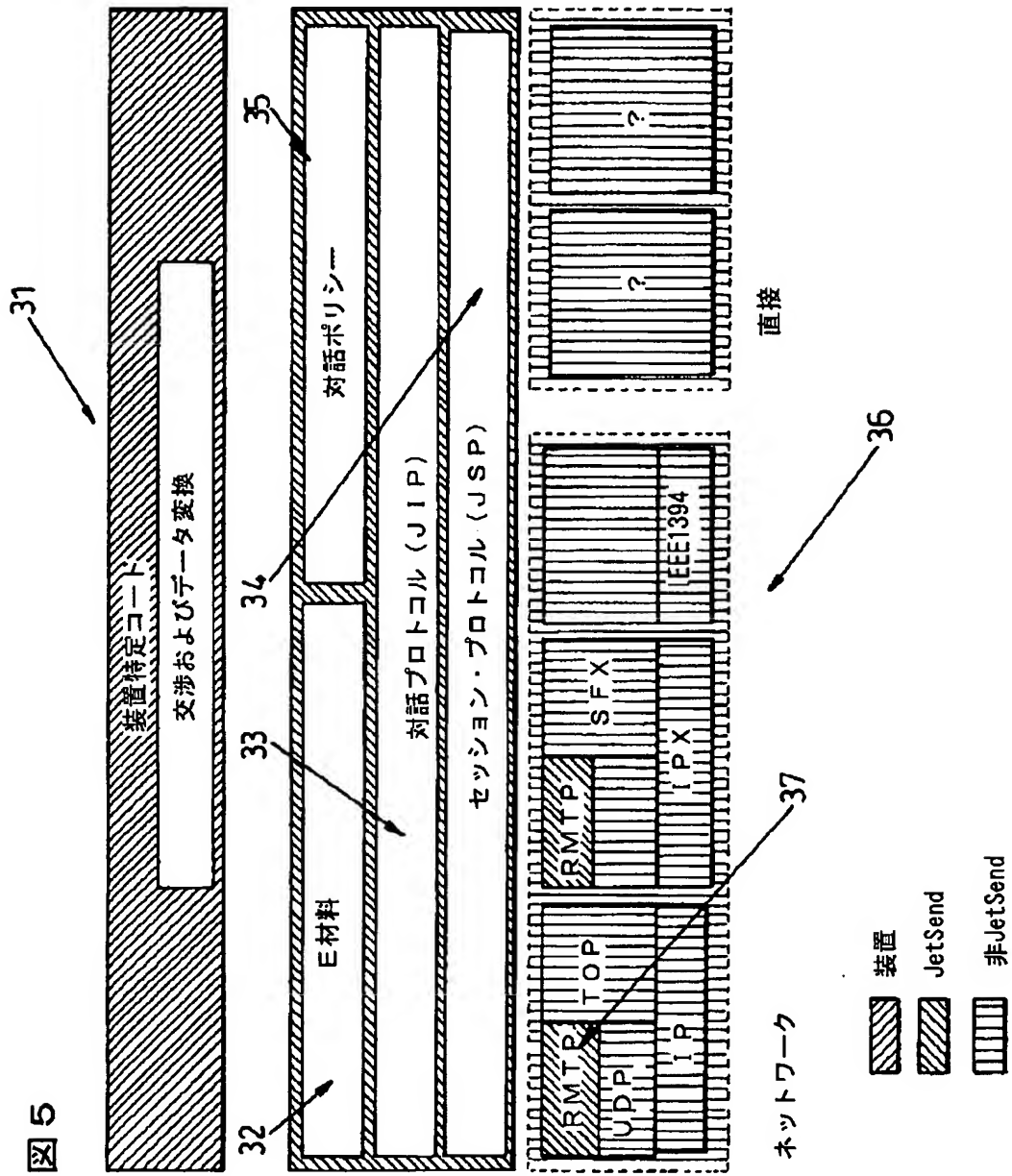


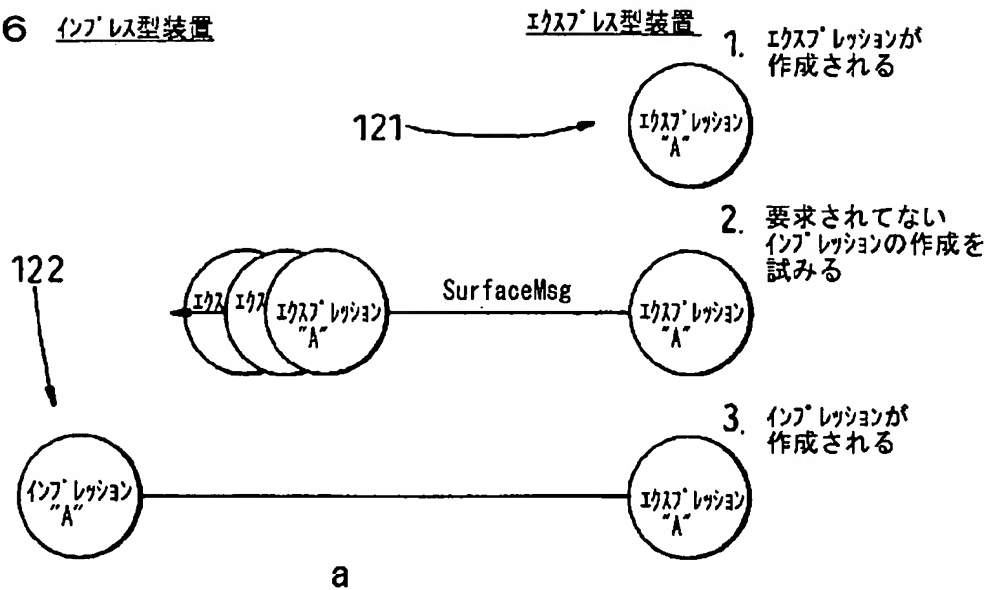
図 4

【図5】



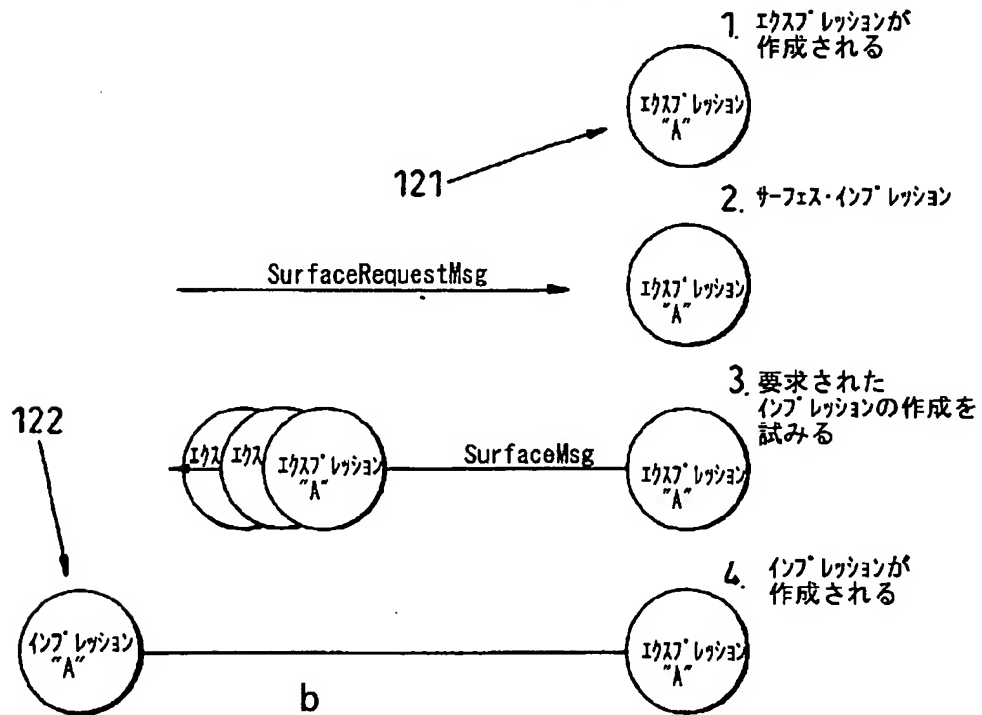
【図6】

図6 インプレス型装置



インプレス型装置

エクプレス型装置



【図6】

図6 インテリ型装置

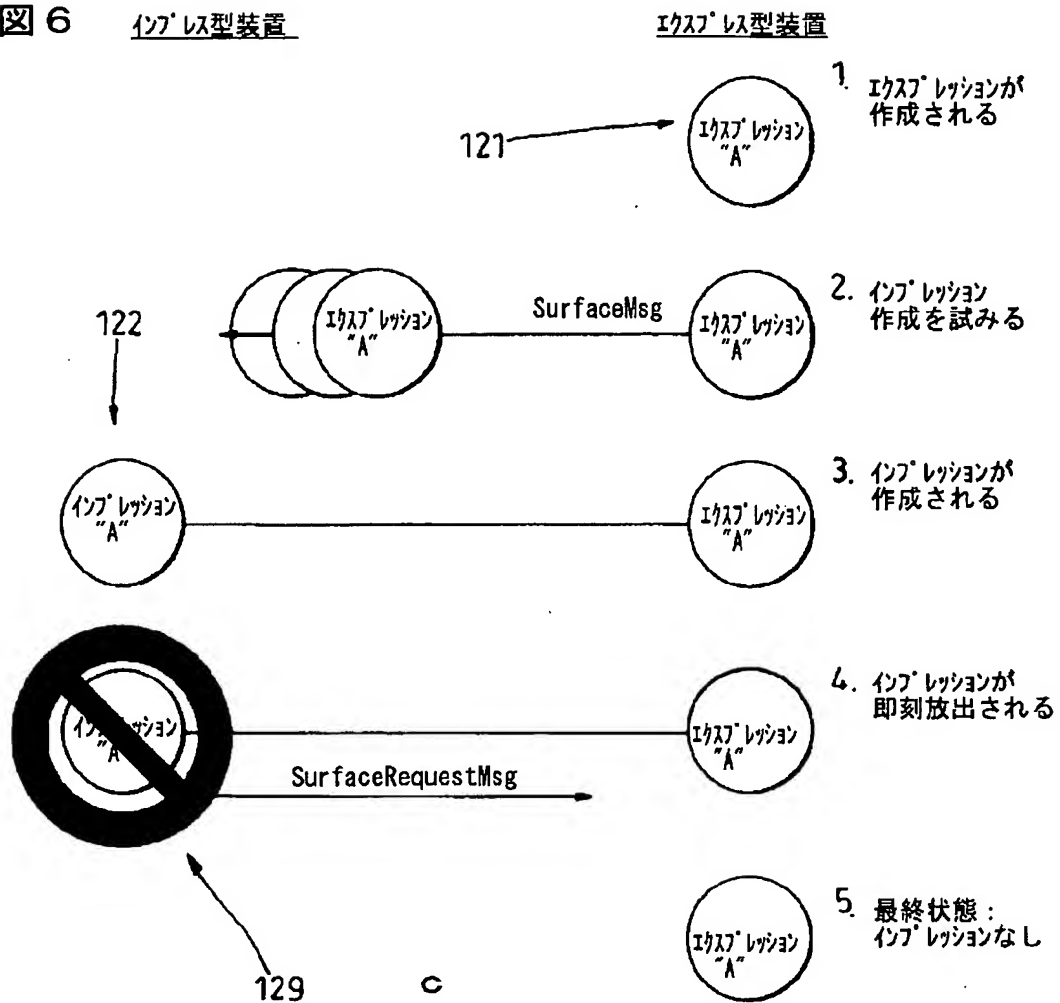
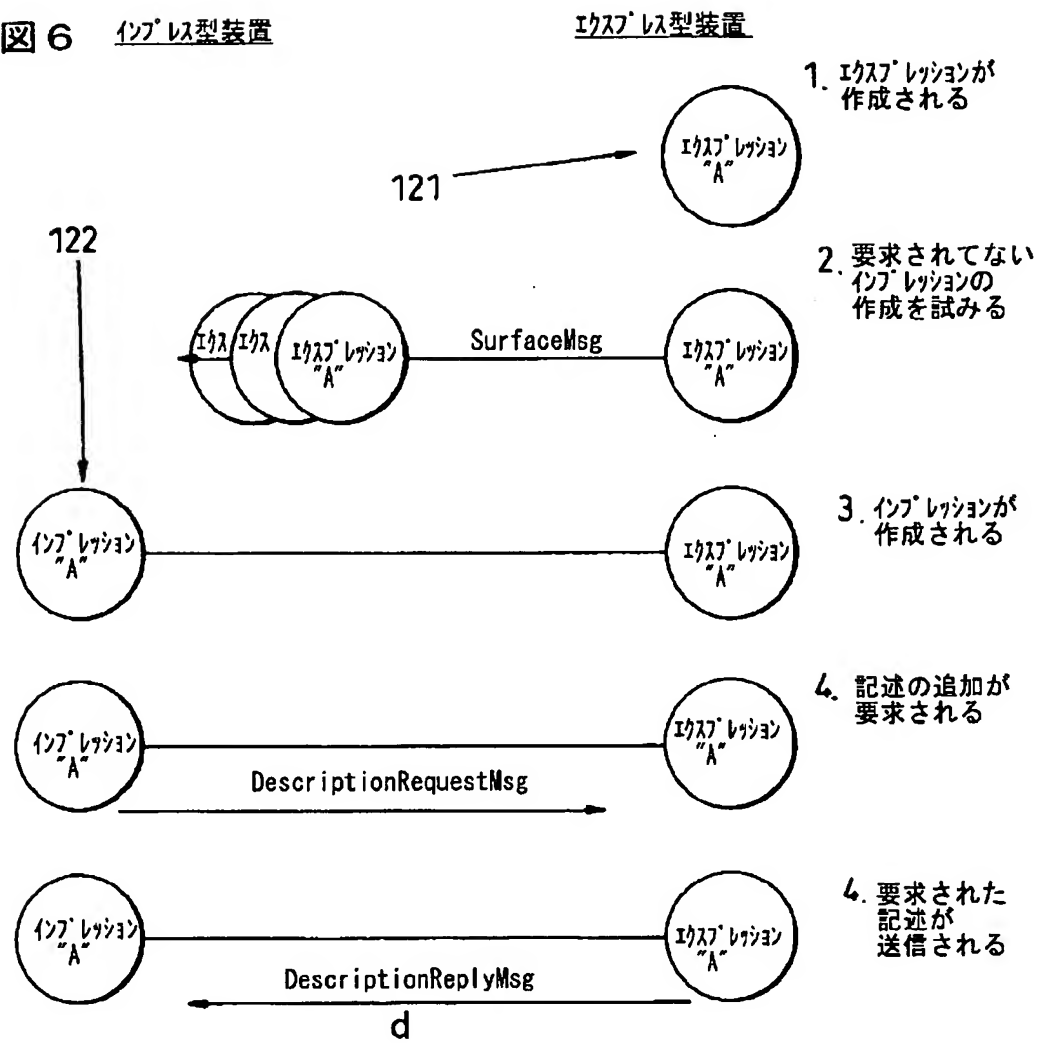


図 6 177° 以型装置

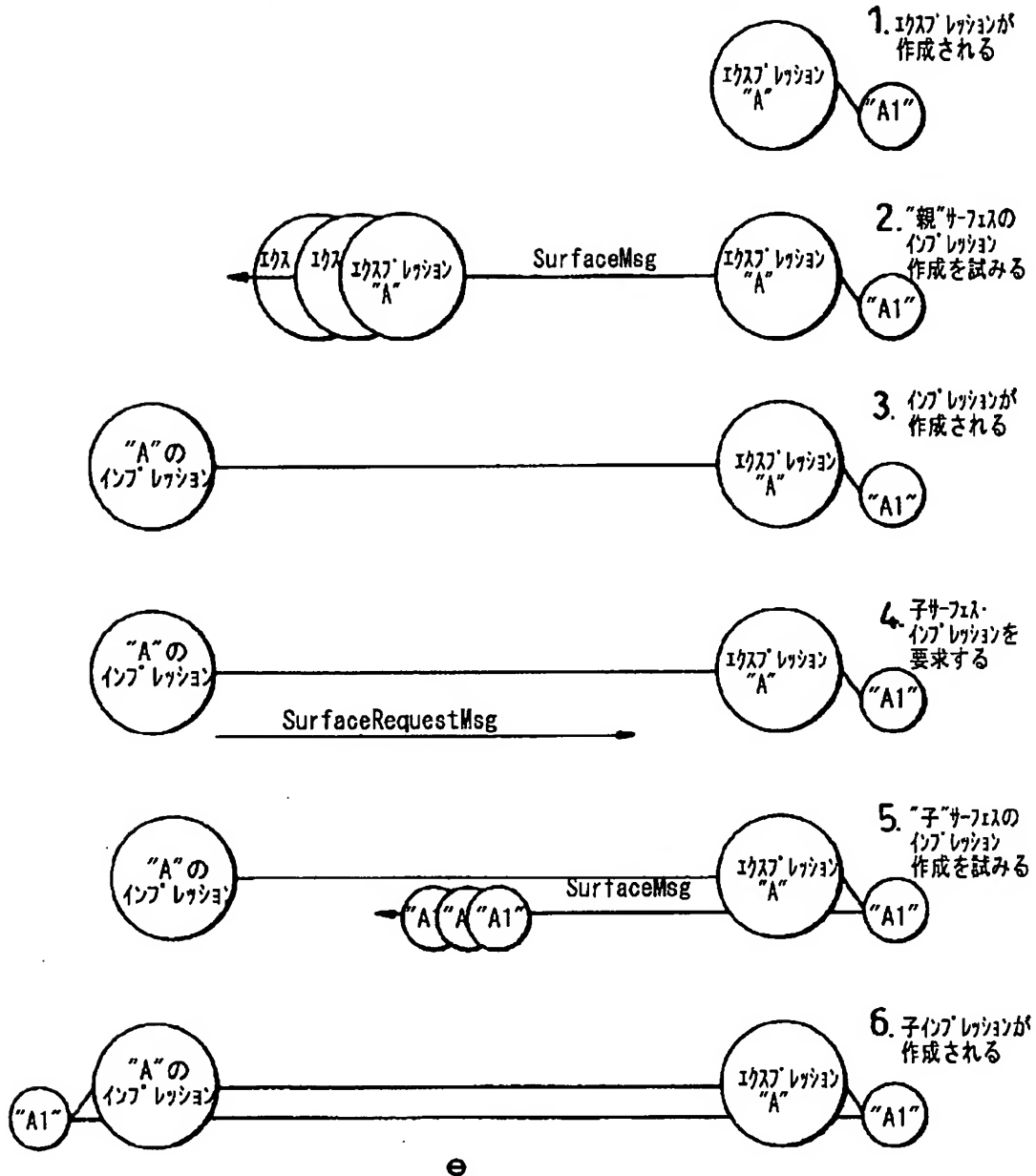


【図6】

図6

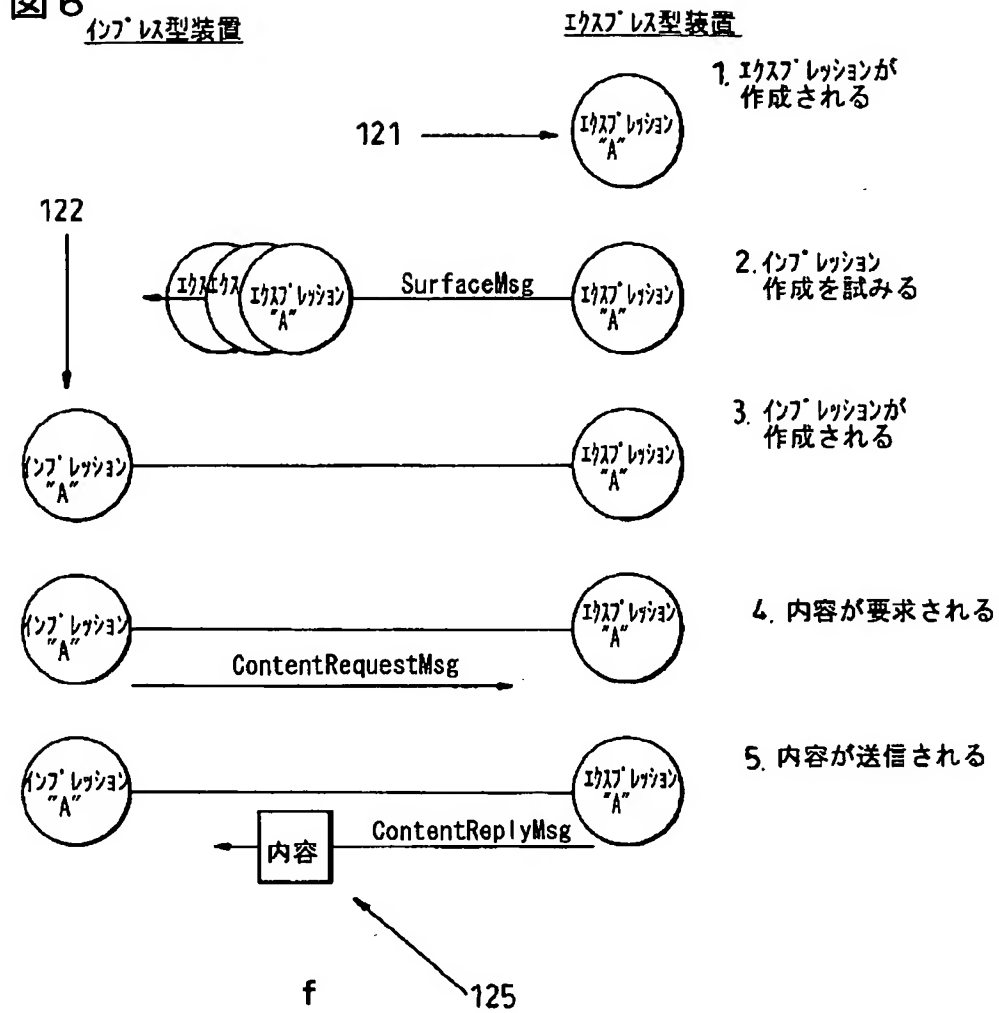
インプレス型装置

エクスプレス型装置



【図6】

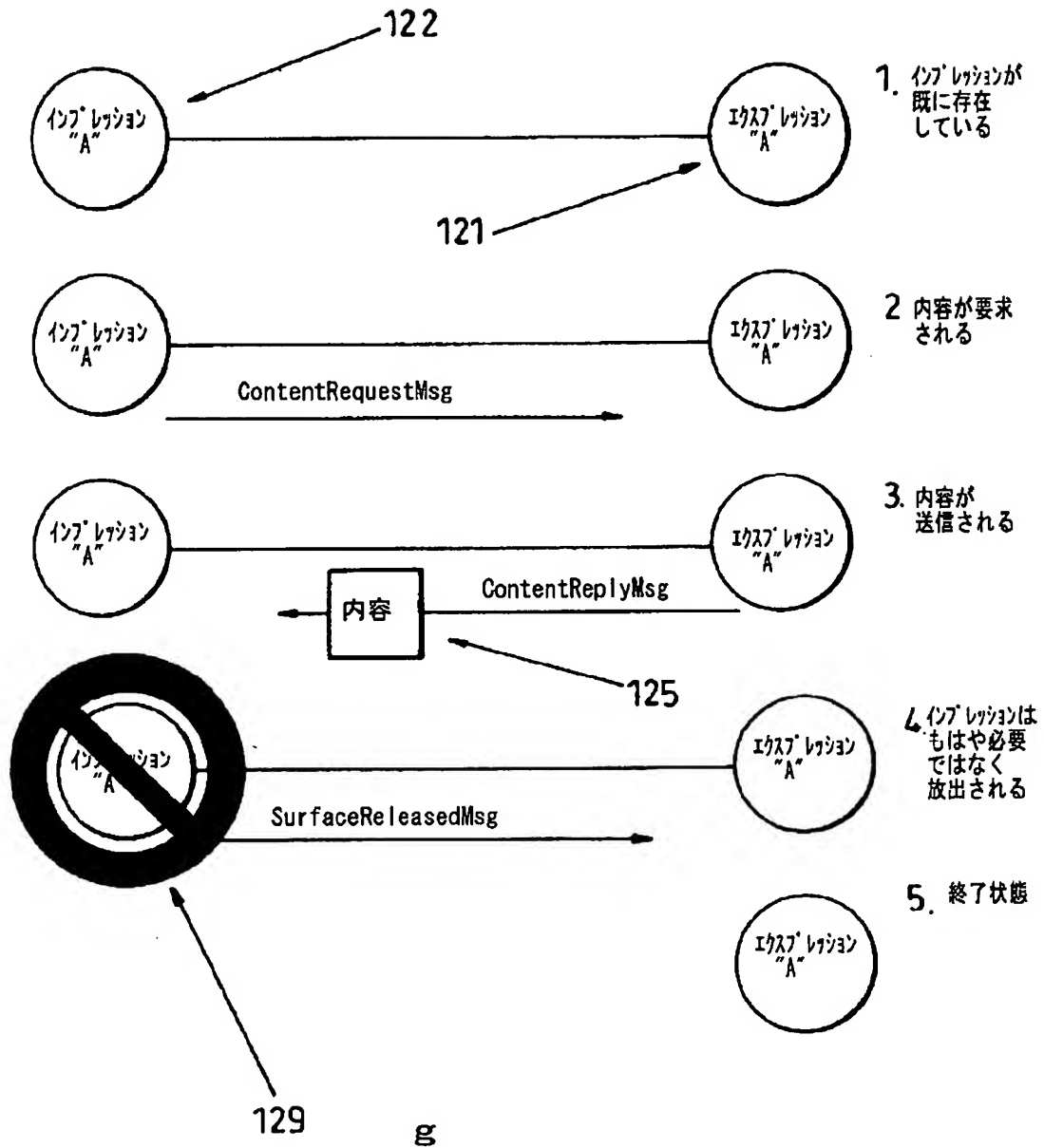
図6



【図6】

図6 インテリス型装置

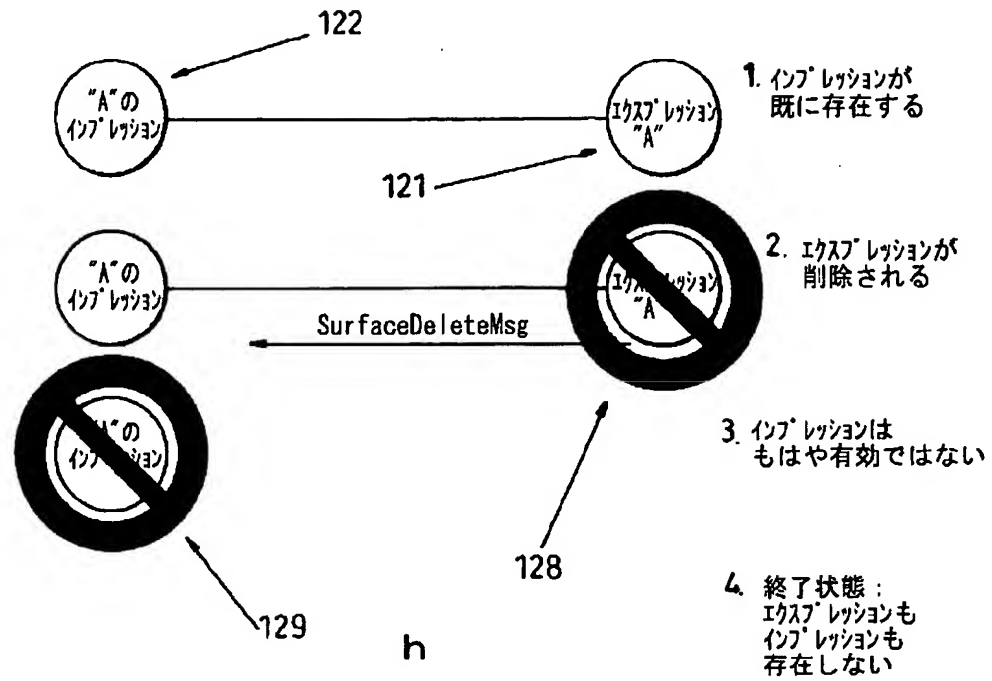
イクスプレス型装置



【図6】

図6 インフレス型装置

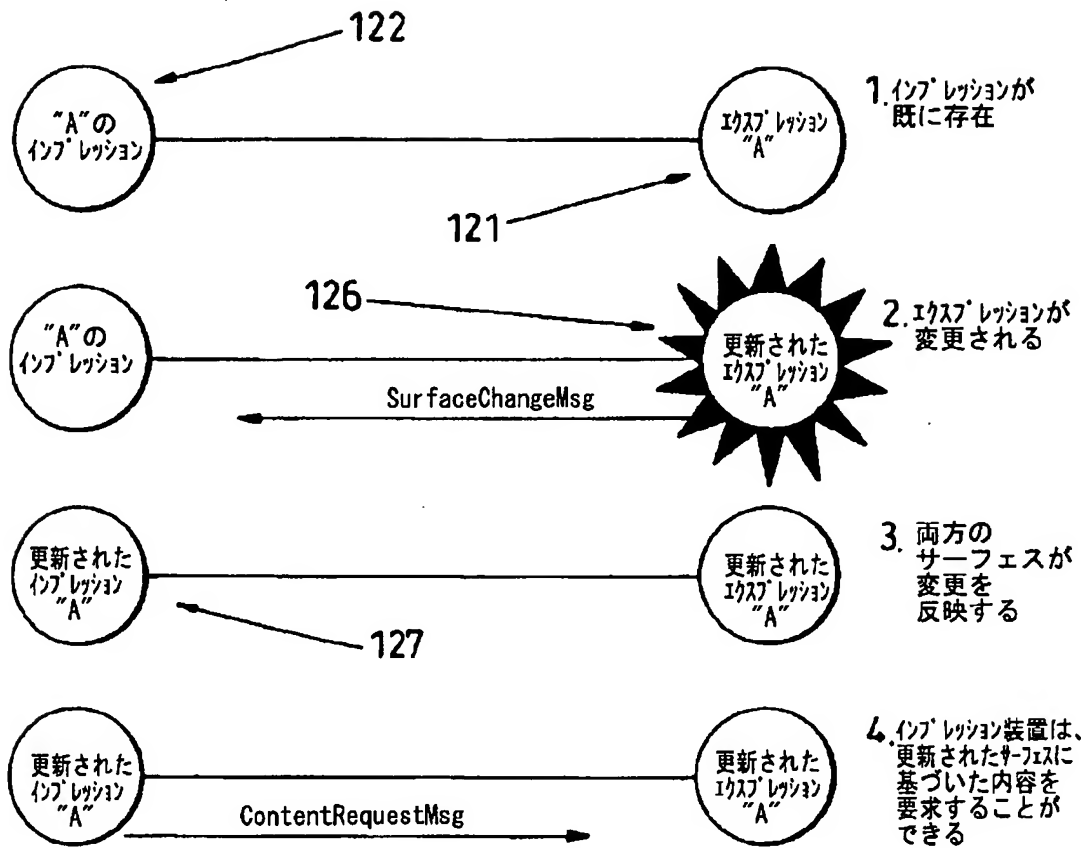
イクスプレ型装置



【図6】

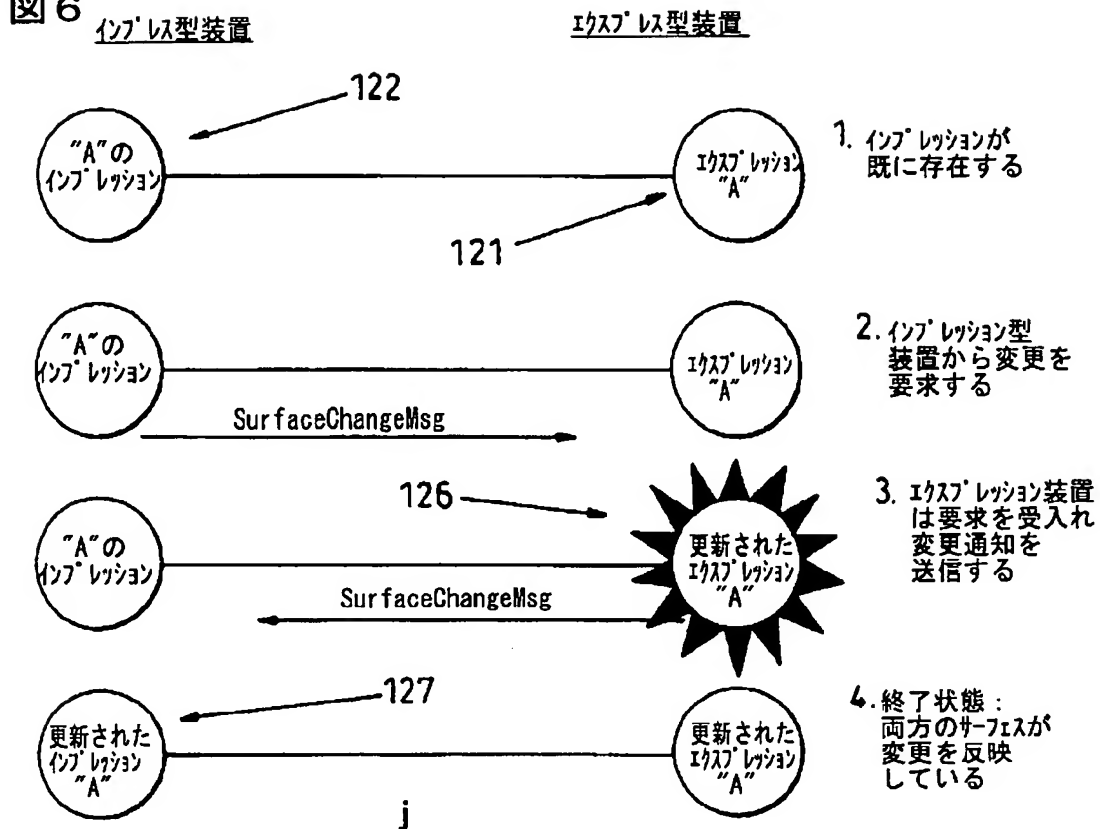
図6 インプレス型装置

イクスプレス型装置



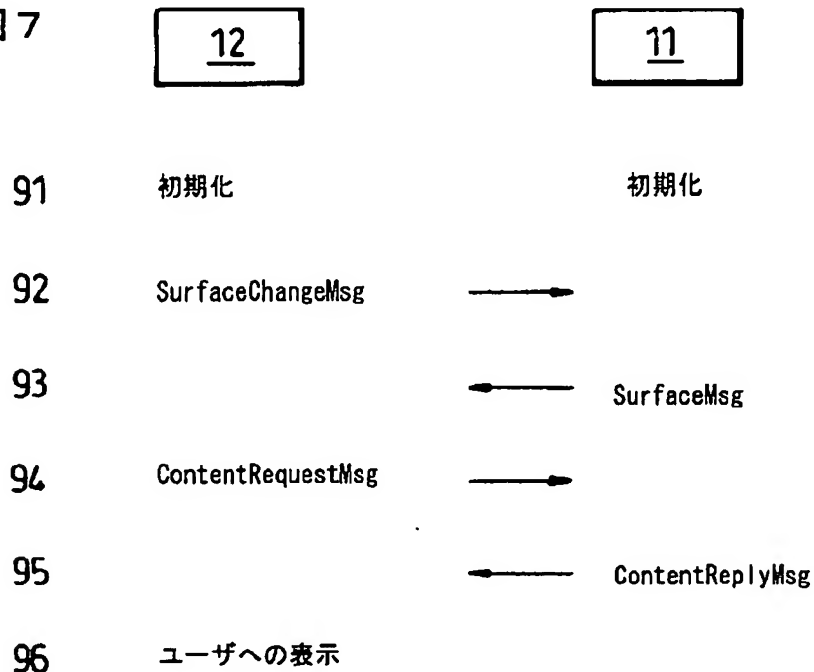
【図6】

図6



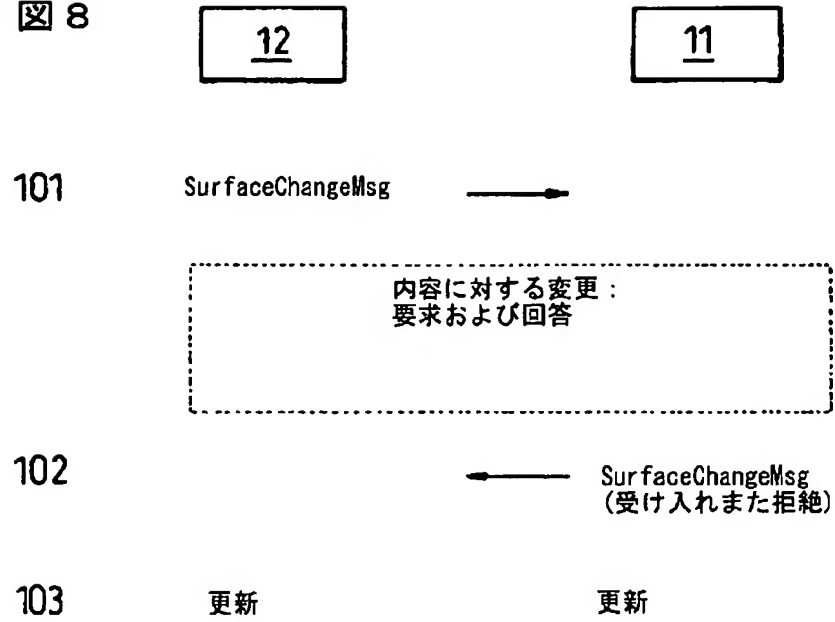
【図7】

図7



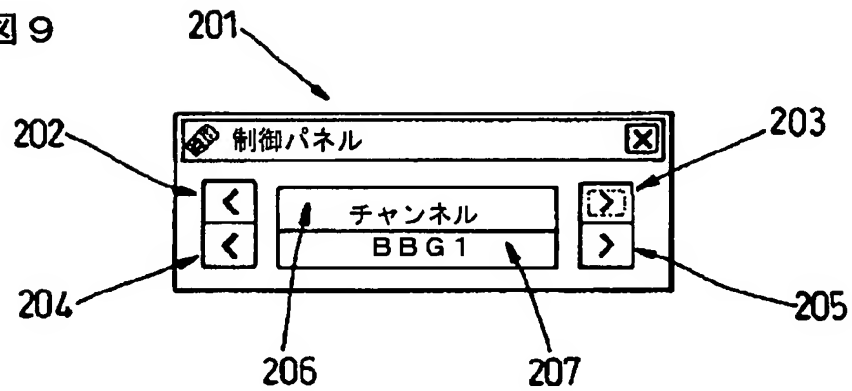
【図8】

図8

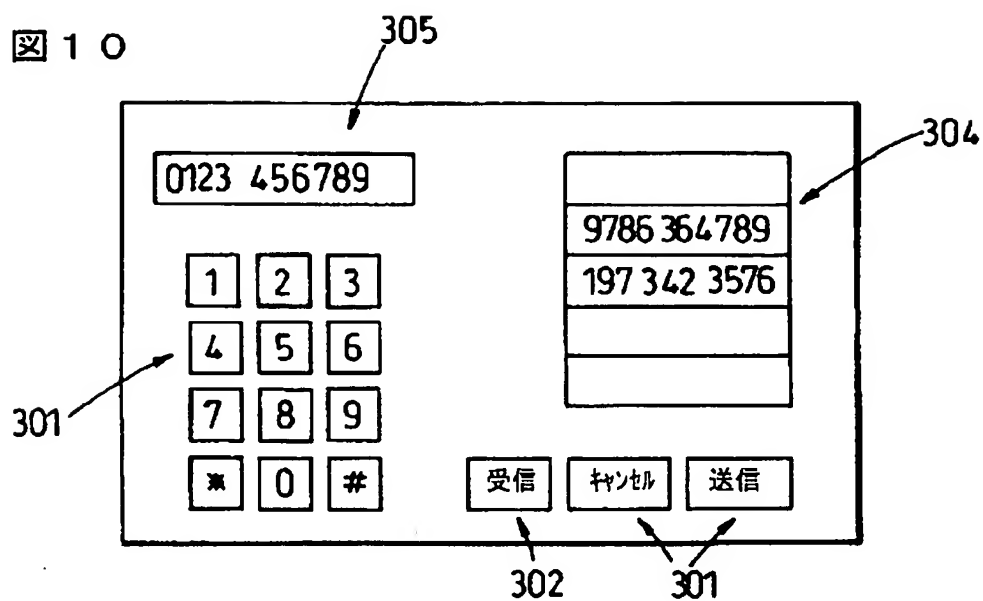


【図9】

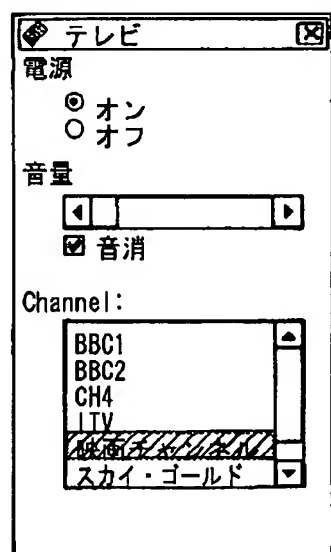
図9



【図10】

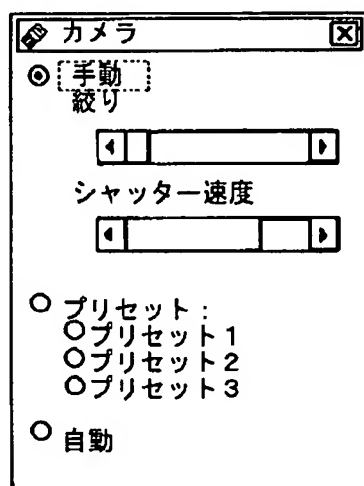


【図10】



a

図10



b

【図11】

図 11

a

電源制御

電源のオンまたはオフ
(クリックして下さい)

b

電源

オン オフ

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/GB 98/01097

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 G06F15/177 G06F13/38		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 6 G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	US 5 317 693 A (CUENOD ET AL) 31 May 1994 see column 7, line 30 - column 8, line 49; figure 9 ---	34-38 1, 2, 4, 29-33
A	WO 94 27223 A (APPLE COMPUTER) 24 November 1994 see page 3, paragraph 1 see page 5, line 10 - line 16 see page 6, line 11 - line 26 ---	1, 2, 4, 29-33
A	EP 0 116 694 A (IBM) 29 August 1984 see page 4, paragraph 1 - page 6, paragraph 1; figure 2 --- -/--	1
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. "Z" document member of the same patent family		
Date of the actual completion of the international search		Date of mailing of the international search report
31 July 1998		20/08/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016		Authorized officer
		G111, S.

INTERNATIONAL SEARCH REPORT

In International Application No.

PCT/GB 98/01097

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	<p>"HP JetSend Technology: Making Device-To-Device Communication Simple"</p> <p>HP PRESS RELEASE,</p> <p>July 1997, pages 1-4, XP002073399</p> <p>http://www2.hp.com/pressrel/jul97/21jul97d.htm</p> <p>Available from Internet on 10 July 1998</p> <p>see page 2, paragraph 7 - page 3, paragraph 2</p> <p>-----</p>	1,2,4, 29-33

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/GB 98/01097

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5317693 A	31-05-1994	NONE	
WO 9427223 A	24-11-1994	US 5634074 A AU 6706594 A	27-05-1997 12-12-1994
EP 116694 A	29-08-1984	HK 8195 A JP 1643460 C JP 3007972 B JP 59154521 A US 4604690 A	27-01-1995 28-02-1992 04-02-1991 03-09-1984 05-08-1986

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY,
DE, DK, ES, FI, FR, GB, GR, IE, I
T, LU, MC, NL, PT, SE), JP, US

(72)発明者 ウィリアムズ・ピーター・マイケル
イギリス、ビーエス20、8 ビーユー、ノー
ス・サマーセット、ポータイスヘッド、メ
ドウズ・クローズ 10、ザ・ホール 4

【要約の続き】

よって決定される。